The Dissertation Committee for Jungup Park
certifies that this is the approved version of the following dissertation:

Data-Driven Modeling and Optimization of Sequential Batch-Continuous Process

Committee:

Thomas F. Edgar, Supervisor

Michael Baldea, Co-Supervisor

Dragan Djurdjanovic

Gary T. Rochelle

Thomas M. Truskett

Data-Driven Modeling and Optimization of Sequential Batch-Continuous Process

by

Jungup Park, B.S.Chem.E., M.S.E.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2016

Dedicated to my family.

Acknowledgments

First and foremost, I would like to thank my advisors, Dr. Thomas F. Edgar and Dr. Michael Baldea for their guidance, trust, encouragement, and wisdom throughout my studies as a graduate student at the University of Texas at Austin. They have led and guided me to achieve my goals while finishing my PhD and taught me many lessons outside the classroom setting, as well. I would also like to extend my gratitude to brilliant professors and engineers who served as my thesis committee members and provided me with fresh perspectives and helpful advices: Dr. Dragan Djurdjanovic, Dr. Gary T. Rochelle, and Dr. Thomas T. Truskett.

I would also like to thank my friends and colleagues from the Edgar and Baldea research groups. They all were great source of knowledge and have made my stay in Austin pleasant and memorable, which I will cherish for the rest of my life. I would like to especially thank Dr. Kody Powell, Dr. Wesley Cole, Dr. Jong Suk Kim, Dr. Ricardo Dunia, Dr. Bo Lu, Dr. Shu Xu, Richard Pattison, Cara Touretzky, Siyun Wang, Matt Walters, Ray Wang, Ankur Kumar, Abigail Ondeck, Krystian Perez, and LTC Corey James.

Lastly, I would not have been where I am without the support of my family. They have always encouraged me and pushed me to become a better student and person. I would like to thank my parents for their unconditional

love and support and my sisters for support and empathy.

Data-Driven Modeling and Optimization of Sequential Batch-Continuous Process

Publication No. _____

Jungup Park, Ph.D.
The University of Texas at Austin, 2016

Supervisors: Thomas F. Edgar
Michael Baldea

Driven by the need to lower capital expenditures and operating costs, as well as by competitive pressure to increase product quality and consistency, modern chemical processes have become increasingly complex. These trends are manifest, on the one hand, in complex equipment configurations and, on the other hand, in a broad array of sensors (and control systems), which generate large quantities of operating data.

Of particular interest is the combination of two traditional routes of chemical processing: batch and continuous. Batch to continuous processes (B2C), which constitute the topic of this dissertation, comprise of a batch section, which is responsible for preparing the materials that are then processed in the continuous section. In addition to merging the modeling, control and optimization approaches related to the batch and continuous operating

paradigms –which are radically different in many aspects– challenges related to analyzing the operation of such processes arise from the multi-phase flow. In particular, we will be considering the case where a particulate solid is suspended in a liquid "carrier", in the batch stage, and the two-phase mixture is conveyed through the continuous stage.

Our explicit goal is to provide a complete operating solution for such processes, starting with the development of meaningful and computationally efficient mathematical models, continuing with a control and fault detection solution, and finally, a production scheduling concept. Owing to process complexity, we reject out of hand the use of first-principles models, which are inevitably high dimensional and computationally expensive, and focus on data-driven approaches instead.

Raw data obtained from chemical industry are subject to noise, equipment malfunction and communication failures and, as such, data recorded in process historian databases may contain outliers and measurement noise. Without proper pretreatment, the accuracy and performance of a model derived from such data may be inadequate. In the next chapter of this dissertation, we address this issue, and evaluate several data outlier removal techniques and filtering methods using actual production data from an industrial B2C system. We also address a specific challenge of B2C systems, that is, synchronizing the timing of the batch data need with the data collected from the continuous section of the process. Variable-wise unfolded data (a typical approach for batch processes) exhibit measurement gaps between the batches;

however, this type of behavior cannot be found in the subsequent continuous section. These data gaps have an impact on data analysis and, in order to address this issue, we provide a method for filling in the missing values. The batch characteristic values are assigned in the gaps to match the data length with the continuous process, a procedure that preserves meaningful process correlations.

Data-driven modeling techniques such as principal component analysis (PCA) and partial least squares (PLS) regression are well-established for modeling batch or continuous processes. In this thesis, we consider them from the perspective of the B2C systems under consideration. Specific challenges that arise during modeling of these systems are related to nonlinearity, which, in turn, is due to multiple operating modes associated with different product types/product grades. In order to deal with this, we propose partitioning the gap-filled data set into subsets using k-means clustering. Using the clustering method, a large data set that reflects multiple operating modes and the associated nonlinearity can be broken down into subsets in which the system exhibits a potentially linear behavior. Also, in order to further increase the model accuracy, the inputs to the model need to be refined. Unrelated variables may corrupt the resulting model by introducing unnecessary noise and irrelevant information. By properly eliminating any uninformative variables, the model performance can be improved along with the interpretability. We use variable selection methods to investigate the model coefficients or variable importance in projection (VIP) values to determine the variables to retain in

the model.

Developing a model to estimate the final product quality poses different challenges. Measuring and quantifying the final product quality online can be limited due to physical and economic constraints. Physically, there are some quantities that cannot be measured due to sensor sizes or surrounding environments. Economically, the offline "lab" measurements may lead to destroying the sample used for the testing. These constraints lead to multiple sampling rates. The process measurements are stored and available continuously in real-time, but the quality measurements have much lower sampling rate. In order to account for this discrepancy, the online process measurements are down-sampled to match the sampling frequency of the lab measurements, and subsequently, soft sensors are can be developed to estimated the final product quality. With the soft sensor in place, the process needs to be optimized to maximize the plant efficiency. Using the real-time optimization, the optimal sequence of manipulated inputs that minimizes the off-spec products are calculated.

In addition, the optimal sequences of setpoints can be calculated by carrying out the scheduling calculation with the process model. Traditionally, the scheduling calculation is carried out without taking the process dynamics into account, which could result in off-spec products if a disturbance is introduced. Incorporating the process dynamics into the scheduling layer poses many different challenges numerically. The proposed time scale bridging model (SBM) is able to capture the input-output behavior of the process while greatly

reducing the computational complexity and time.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Batch to Continuous (B2C) Processes

A sequential batch-continuous (B2C) process is a hybrid system where the raw material is initially processed in a batch fashion upstream, and the processed material is converted to the final product in a continuous fashion downstream (Figure 1.1). In order to understand and successfully operate the process, mathematical modeling, control, and optimization of the process need to be carried out, in that respective order (Figure 1.2). First, the operating goal needs to be defined, which comes from the business and organizational interest. After setting a realistic goal, historical data need to be collected in order to develop a model. Also, the training data need to have enough information (rather than just containing noise). Determining the training data set is important as this data set defines the scope and the range of the normal operation. The raw data without any treatment can contaminate with noise and distort the model, and could lead to reduction in model accuracy [1]. After properly cleaning the data, a proper model form needs to be defined and its model parameters need to be calculated. The developed model needs to be validated using a testing data set prior to taking any further steps. This developed model can be implemented and used for monitoring, which is

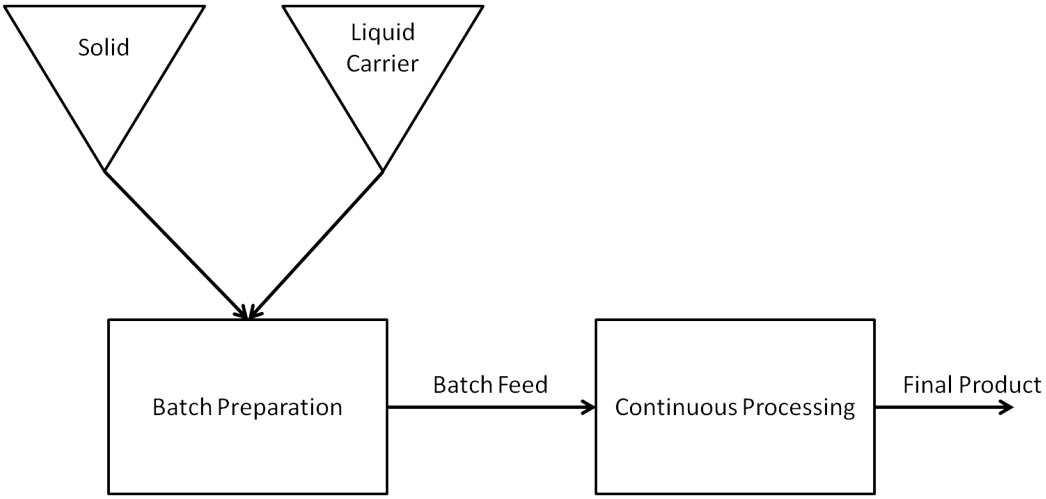

Figure 1.1: Schematic of Sequential Batch-Continuous System

evaluating the current status of the operation. To ensure the safe and profitable production, the developed model can be used to devise a control strategy. After having a closed-loop control system in place, the process operation can be optimized in order to maximize the profit while meeting all the process constraints and product demands.

1.2 Dissertation Outline

In this dissertation, the mathematical methods and algorithms required to achieve the road map shown in Figure 1.2 are discussed. It is organized as follows:

Chapter 2 focuses on the data cleaning and batch data alignment for sequential batch-continuous process. The first section primarily focuses on data

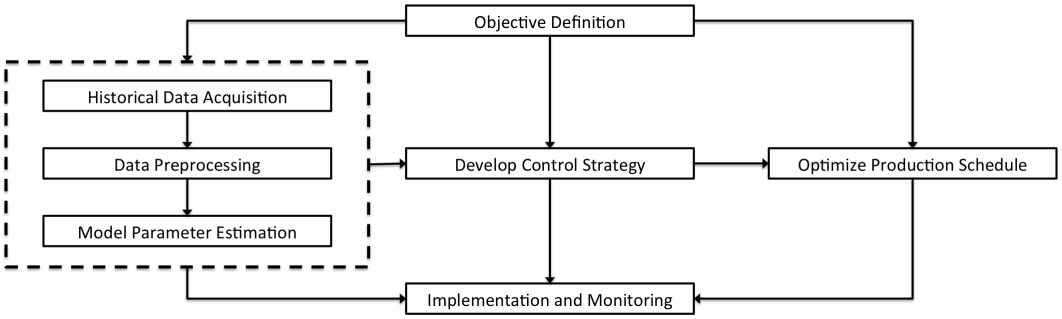

Figure 1.2: Flow Diagram of Data-Driven Modeling, Control, Optimization and Implementation

cleaning methods. It reviews and compares the existing data cleaning methods such as outlier removal and filtering. The second section discusses the batch data alignment to the continuous data for sequential batch-continuous process, which is shown in Figure 1.1. Variable-wise unfolded batch data exhibits gaps between batches and these gaps in data can pose a specific challenge for sequential batch-continuous process. These gaps cannot be simply discarded because they contain information for correlating the batch and continuous processes. In this section, a novel method to assign batch characteristic variable to the gaps between batches is introduced in order to establish correlation in sequential batch-continuous process.

Chapter 3 introduces data-driven modeling techniques. The prevalent and powerful methods such as PCA and PLS are described in detail. It also shows how these methods can be applied to sequential batch-continuous process for monitoring purposes. In order to deal with multiple operating modes and process nonlinearity, prior to developing data-driven models, the data set

is partitioned using clustering method. Lastly, the section gives an overview of variable selection methods for PLS models. The introduced methods are compared to determine which method suits the sequential batch-continuous process.

Chapter 4 shows how to incorporate the developed model to operate the process optimally. The first section discusses the development of a soft sensor that computes the real-time evaluation of a quality variable with low measurement frequency. The second section proposes different real-time optimization formulations to calculate the optimal sequence of inputs in order to produce in-spec products at a higher rate.

Chapter 5 integrates modeling and control with scheduling. The process with controller in place needs to have predefined setpoints, but determining the sequence of setpoints for the plant still remains as an open-ended question. In order to find the sequence of setpoints that satisfy the product demands and maximize product profit, a scheduling problem needs to be solved. Different solution approaches for integrating model under control and scheduling are introduced alongside with the extensive review of current literatures. In order to incorporate process dynamics while reducing the computational complexity, a novel framework called the time scale bridging model is introduced. This method is compared to other solution approaches.

Lastly, Chapter 6 draws the conclusions and summarizes all the work presented in this dissertation. Also, some directions for future work are laid out.

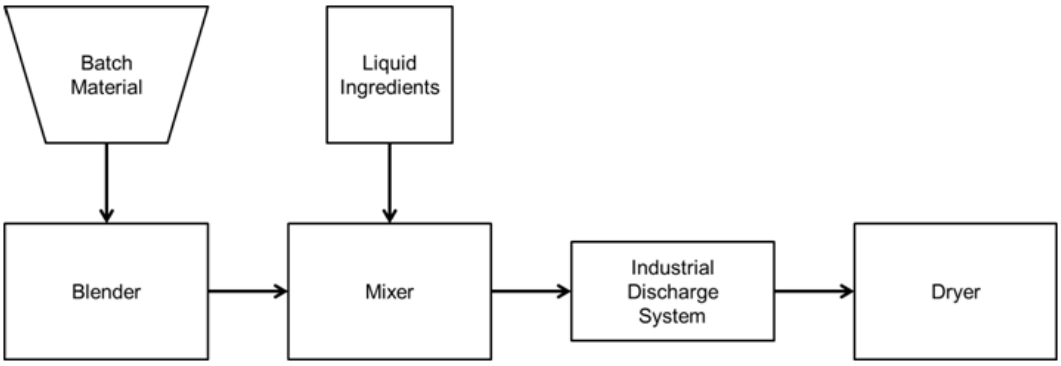

Figure 1.3: Schematic of Industrial Two-Phase Material Processing System

The methods and algorithms discussed in this thesis have been tested with the actual plant data from an industrial sequential batch-continuous process. This system exhibits two-phase suspension (solid particles in liquid carriers) with upstream batch preparation followed by downstream continuous processing to finalize the product. The schematic of the two-phase material processing system is shown in Figure 1.3. In this particular system, the initial raw material (solid) is blended in a batch fashion to a desired property. Subsequently, this batch material mixed with a liquid carrier. This two-phase mixture is, subsequently, fed to the continuously operating processing system in a batch fashion. The continuous processing system pushes the material to the exit to obtain the desired shape, form and material properties.

Chapter 2

Data Cleaning and Batch Data Alignment

2.1 Introduction

Complex chemical processes can involve both batch and continuous stages. Raw materials and other ingredients are initially processed batch-wise, prior to being fed to a processing line that operates continuously. This type of hybrid processes can be found in many different industries such as pharmaceutical, food processing, polymer processing, and semiconductor manufacturing [16, 85, 107, 6, 18]. The operating conditions and operating performance of both the batch and the continuous stages have an impact on the final product quality. Such sequential batch-continuous processes pose specific analysis and control challenges. The batch side of the process operation is carried out periodically at specified time intervals. After each operating instance, the batch product is fed to the continuous production flux. Empirical evidence suggests that this mode of operation leads to a deterioration of the causal relation between the properties of the batch product and the quality of the product of the continuous process. This is further complicated by the time delay that is inherently introduced by the continuous stage of the process between the completion of the batch stage and any quality measurements obtained from the final product. The batch and continuous parts have impact on

the final product quality; in order to implement a control strategy to obtain acceptable final products, it is essential to establish a relationship between the batch and continuous parts. In this work, a specific industrial sequential batch-continuous process was used for modeling, testing and implementation; however, the methods can be applied to many other hybrid processes that rely on both batch and continuous processing.

The industrial B2C process involves several stages (Figure 1.3). In the first stage, raw material is blended in batches through sequential steps. Liquids are added to the batch and mixed well before being fed to the continuous section, which is a distributed parameter system: the material is transported through a tubular apparatus where it is subject to compression and shear, then discharged through a die to obtain a product of desired cross-section and geometry. Temperature and pressure measurements are taken at various locations along the system. This type of processing is present in many industries; familiar examples include the production of pasta, aluminum bars with complex profiles used, e.g., in the manufacturing of window and door frames, and some of the uses of the familiar children's toy, Play-Doh ®[85, 107, 18].

The objective of this process is to maintain the final product quality within specifications despite variations and disturbances occurring in upstream process units at unpredictable times. One of the cause of variations is the change in quality of raw materials. Raw material blend quality may vary over long periods of time due to seasonality, daily temperature and humidity variations, storage condition changes or raw material supplier changes [107]. Such

shifts in raw materials cause changes in properties of the batch and produce undesired variability in product quality. A common practice to avoid this issue is to measure some batch properties and make changes in the continuous part of the process so that the undesired variability can be eliminated. This approach helps reduce variability caused by upstream disturbances; however, it may be impossible or difficult to obtain a meaningful quality measurement of the material as it is transferred between the upstream batch process and the downstream continuous process. In certain cases, a manual feedback loop, in which the operators collect samples from the continuous process, takes quality measurements and makes decisions on changing manipulated variables (which are discussed later), is implemented. Measurement errors, measurement bias due to operator shift changes, destructive loss of product for testing and slow response of the feedback loop are a few disadvantages of the current approach.

In light of the above, some of the salient challenges involved in the data-driven modeling and analysis of B2C processes are:

1. **Gap in Batch Data**

   Variable-wise unfolded batch data sets contain gaps in measurement between batches (Figure 2.1). Typically in a sequential batch-continuous process, the batch cycle is predetermined to avoid any disruption in continuous process in order to maximize profit and minimize downtime(Figure 2.2). These gaps in data set pose a challenge in establishing a relationship between the batch and continuous processes. As shown in

Figures 2.1 - 2.2, even though there are gaps in measurements between batches, no such behavior is observed in continuous side. The length of batch data measurements is significantly lower than the length of the continuous measurements; however, physically, there is no downtime in continuous production because a subsequent batch is prepared and gets fed to the continuous process before the previous batch has completely exited the continuous process. The treatment of these gaps will likely have a strong impact on the analysis of the data and the interpretation of the results. If the data collected from the continuous process during these "gaps" are simply discarded, meaningful information concerning the continuous process and the impact of the batch process upstream (and in particular, the impact of *past* batches) may be lost. Leaving the data gaps as-is is bound to be interpreted as a process failure, with undesirable consequences on any post-analysis uses of the resulting models.

2. **Multiple Steady-states** A complex chemical process that produces different grades or types of product by varying operating conditions (and the corresponding controller setpoints) or inputs, features multiple operation modes [88], each typically having a different operating steady state. Moreover, processes of practical interest have a nonlinear behavior and, as such, it is difficult to capture the characteristics of the process over the entire operating space using a single linear model. In an ideal case, a global model can be developed that takes the nonlinearity and the complexity of the system dynamics into account, which

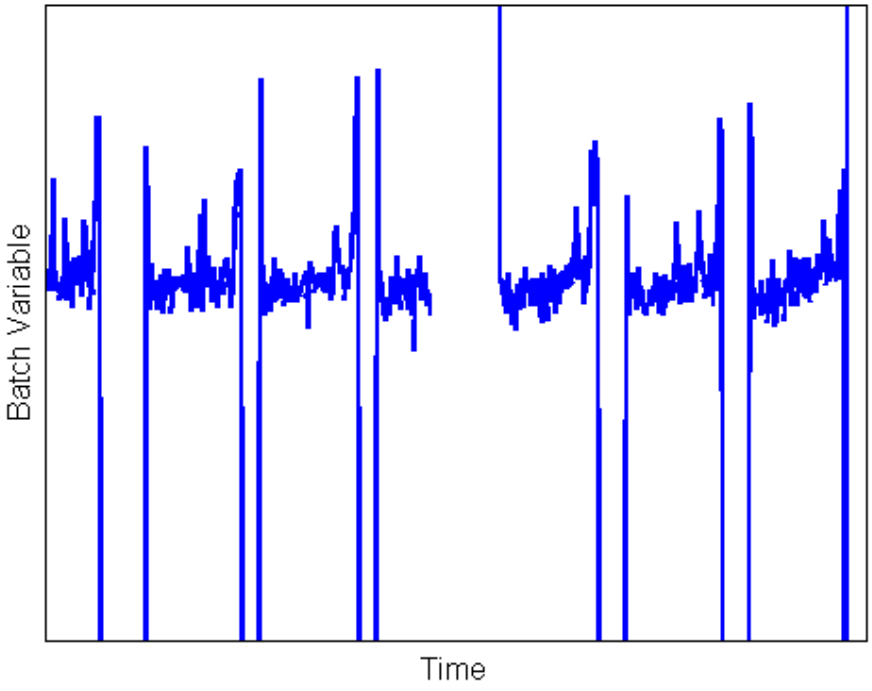

Figure 2.1: Plot of Process Measurement in Batch Process

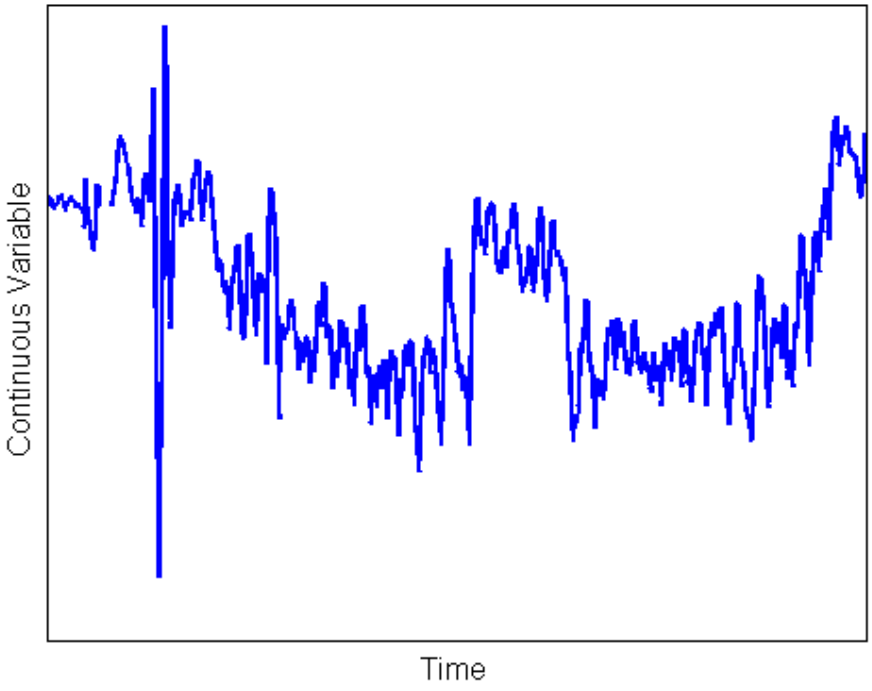

Figure 2.2: Plot of Process Measurement in Continuous Process

can be utilized for all operating conditions; however, this is a difficult task in practice. Such models are inevitably hard to develop, expensive to maintain and unwieldy to use for optimization and control. Instead, multiple *linear* models can be developed for a subset of operating regions. In this case, there are other challenges, including partitioning the operating space (and/or the data set) and establishing a meaningful (and practical) number of models to develop.

3. **Nonlinearity and Dynamics** Batch processes do not operate at a steady-state; their operation is inherently dynamic. This characteristic is also reflected in the behavior of the downstream, continuous section of the process. Intuitively, the continuous section will exhibit a periodic nature, with the characteristics of the periodic behavior related to the timing of the batch operations. Moreover, variations or disturbances in the upstream batch section are likely to result in shifts in the downstream continuous process.

   Additional modeling challenges arise from the complexity of the material processed in the system, which is in many cases a two-phase fluid whereby solid particles are suspended in a liquid carrier.

4. **Measurement Noise and Bias** "Raw" chemical process measurements (i.e., a direct recording of a sensor reading) tend to exhibit high noise and contain many outliers, as shown in Figures 2.1 - 2.2 [1, 22, 25, 117, 38]. These characteristics could have negative impact on model development,

and need to be addressed properly [63, 80, 129]. Also, lab measurements such as operator-made offline measurements could be biased due to human error. Multivariate analysis such as PCA can handle some level of noise, but can be improved in accuracy and interpretability, if the input data set is cleaner [26, 111, 108].

5. **Lack of Intermediate Quality Measurement** In sequential B2C process, it is often the case that no measurements or quality testing of the product conveyed between the batch and continuous sections are taken. Rather, the focus is on the end-point product quality of the continuous section. Without securing measurements of a (set of) intermediate quality variable(s) that can be used to link the evolution of the batch and continuous sections, the difficulty of establishing a causal relationship between events occurring at the batch stage, and fluctuations observed in the continuous stage, increases.

As a separate issue, batch data are three dimensional, with the three dimensions being the variable list, the sample time, and the batch number. In order to correlate such data with data obtained from the downstream continuous process, the data must be "unfolded" into a two-dimensional data matrix. The usually large number of measurements/sensors used to generate the data, and high sampling frequencies, cause this matrix to be quite high dimensional; typical industrial data sets used in the examples presented in this thesis comprise millions of rows and hundreds of columns. The abundance of

data does not directly and immediately result in an abundance of process *information*, and the size of these data sets may in fact have a deleterious effect on extracting meaningful and actionable process information. [129, 63, 80].

Existing multivariate statistical techniques can be applied to reduce the dimensionality of process data sets. In the batch realm, multi-way principal component analysis (MPCA) [94, 127, 131, 75], dynamic PCA [72, 78, 132], etc. have been used. PCA and PLS have been successfully applied to continuous processes [59, 23, 24, 57, 66, 68, 95, 105]. Nevertheless, it is not clear at present what approach should be taken for the B2C processes, which, as emphasized above, have some of the characteristics of both batch and continuous systems. As an additional challenge, the high dimensionality of the data can lead to numerical issues in the development of latent-variable models, such as overfitting and ill-conditioned matrix operations [26, 111, 108].

2.2 Data Preprocessing

A typical chemical process measurement contains measurement noise, transients, and outliers which need to be dealt with prior to undertaking any model-building effort [81]. Data "cleaning" techniques must, however, be applied with care. For example outlier removal methods which use the mean and standard deviation to establish confidence intervals and thresholds for removing outlying data points can introduce time delays in the remaining data, an issue not encountered when applying techniques based on median and median absolute deviation (MAD). In this section, we discuss available methods for

outlier removal from time series data such as those collected from chemical processes; additionally, we focus on literature developments concerning filtering high-frequency noise that is inherently present in industrial data.

2.2.1 Outlier Removal

2.2.1.1 Motivation

Outliers are defined as [14]:

> "a observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data."

In the chemical industries, outliers are associated, e.g., with errors in data transmission or transcription, contaminated samples, or malfunction in sensors. Outliers must be properly treated and removed prior to moving on to subsequent steps such as model identification or data analysis, as they can heavily (and negatively) affect the these steps.

2.2.1.2 Methods

Outlier removal methods fall into two broad categories: *univariate*, which consider the variables in a multi-variable data set (and the corresponding outliers) individually, and *multivariate*, which take into consideration the interactions between the variables in a data set when performing outlier removal.

2.2.1.2.1 Univariate Outlier Removal Methods The univariate data-driven methods rely on defining a confidence interval. A significance level (α), $0 < \alpha < 1$ has to be chosen, then the α-confidence interval of the $N(\mu, \sigma^2)$ distribution is defined as follows:

$$out(\alpha, \mu, \sigma^2) = \left\{x : |x - \mu| > z_{1-\alpha/2}\sigma\right\} \tag{2.1}$$

z_q is the quantile function for the cumulative distribution function, Φ, where $\Phi(x) = q$. Without knowing the actual distribution $\Phi(x)$, the normal distribution can be used as the target distribution, but this definition is not limited and can be extended to any unimodal symmetric distribution with positive probability density function.

1. **Extreme Studentized Deviate (ESD) Identifier** The ESD identifier is a frequently-used method; outliers are detected as follows [101]:

$$CI = \mu \pm \sigma\phi(N, \alpha) \tag{2.2}$$

After calculating the mean and the standard deviation, the confidence interval can be defined. α is the significance level and $\phi(N, \alpha)$ is the probability density function (PDF) of Gaussian distribution for the given N and α. After the confidence interval is defined, the outliers are detected as any points that lie outside the calculated confidence interval. In this method, the number of samples (N) and significance level (α) serve as

two degrees of freedom to adjust the aggressiveness of the identifier. Similar to ESD identifier, a widely known 3σ method can be applied. Instead of using the PDF of Gaussian distribution, a constant value of 3 is used in Equation 2.2. These methods only apply to independent and identically distributed (i.i.d.) data sets that follows Gaussian distribution [14, 100]. Implementing the ESD identifier online is straightforward. The statistical quantities such as μ and σ need to be predetermined from a historical data set. Using these values, the incoming data are compared to the confidence interval shown in Equation 2.2, and the samples that lie outside are considered outliers and removed.

2. **Hampel Identifier** The Hampel identifier is similar to the ESD identifier, but instead of using the mean and standard deviation, it utilizes the median and median absolute deviation (MAD), which is defined in Equation 2.4. The confidence intervals are defined as follows [101]:

$$CI = med \pm MAD\phi(N, \alpha) \tag{2.3}$$

$$MAD = med_i(|X_i - med_j(X_j)|) \tag{2.4}$$

The difference between the identifiers comes from mean and median. The Hampel identifier is more aggressive and identifies short-lived outliers better than ESD identifier because the median is less influenced by

outliers than the mean. Similarly, the number of samples (N) and significance level (α) can be used to adjust the aggressiveness of the identifier. Online implementation of Hampel identifier is similar to ESD identifier. Once *med* and MAD are defined, Equation 2.3 can be used to determine if the incoming sample is an outlier.

3. **Quartile-based Identifier** Another identifier utilizes interquartile range (IQR), where [101]

$$Q = x_U - x_L \tag{2.5}$$

x_L denotes the lower quartile, $x_{(0.25)}$, and x_U denotes the upper quartile, $x_{(0.75)}$.

Utilizing the IQR, the outlier range for symmetric distribution can be defined as follows [101]:

$$CI = \frac{x_U + x_L}{2} \pm 2Q \tag{2.6}$$

where, the first term $\frac{x_U+x_L}{2}$ is the median. This identifier, however, is less effective than the Hampel identifier, owing to its confidence interval being typically wider. The advantage of the Quartile-based identifier is that unlike the two identifier mentioned above, it can be applied to asymmetric data distributions simply by using the a modified confidence interval [101]:

$$CI_L = x_U + 1.5Q = 2.5x_U - 1.5x_L \tag{2.7}$$

$$CI_U = x_L - 1.5Q = 2.5x_L - 1.5x_U \tag{2.8}$$

Also, the quartiles can be used to create a boxplot, which can be a graphical illustration of the range of the identifier. Similar to the previous cases, as long as the confidence interval is predefined from historical data, this method can be easily implemented online using either Equation 2.6 or Equation 2.7.

2.2.1.2.2 Multivariate Outlier Removal Methods

1. **Hotelling's T^2 and Q-statistics** Unlike the methods mentioned above, the Hotelling's T^2 and Q-statistics can handle multivariate outliers [86, 67, 28, 103]. PCA utilizes statistical values such as mean and standard deviation/covariance, which can only be applied to Gaussian distributed data set. Due to this limitation, a data set that exhibits nonlinearity cannot be treated using these latent-variable methods. Also, both the Hotelling's T^2 and Q-statistics must follow multivariate normal distribution assumption. There have been other variations of PCA developed such as Dynamic PCA (DPCA) [72, 110, 78] and kernel PCA (KPCA) [113, 29, 30] in order to handle dynamics and nonlinearity, respectively. More details about PCA will be discussed in the following chapter, and a brief overview is presented below in the context of outlier removal.

First, PCA has to be performed to decompose the data set into linearly independent principal components.

$$X = TP^T + E \tag{2.9}$$

In Equation 2.9, X is the original $m \times n$ data matrix, having m measurements/observations of n variables, T is the "score" matrix, and P is the "loading" matrix, and lastly, E is the residual that is not captured by the projected data. Hotelling's T^2 statistic is computed as[86, 67, 28, 103]:

$$T_i^2 = t_i \lambda^{-1} t_i^T \tag{2.10}$$

where, t_i is the ith row of the score matrix (T), and λ is a diagonal containing the eigenvalues. The Q-statistic is defined as follows [86, 67, 28, 103]:

$$Q_i = e_i e_i^T \tag{2.11}$$

where, e_i is the ith row of the residual matrix (E). In essence, T^2 considers the variations in the principal component subspace (within the model), and Q-statistic measures the magnitude of the sample projection on the residual subspace (outside the model).

Using the Hotelling's T^2 and Q-statistics for (online) outlier detection, the loading matrix (P) from the training data set needs to be utilized.

The new score T_{new} for the incoming data set X_{new} (or data sample x_{new}) can be calculated using the matrix operation shown in Equation 2.12. Similarly, the error matrix E_{new} for the incoming data set can be computed using Equation 2.13. Once T_{new} and E_{new} are obtained, Equations 2.10 and 2.11 can be used for computing the Hotelling's T^2 and Q-statistics.

$$T_{new} = X_{new}P \tag{2.12}$$

$$E_{new} = X_{new} - T_{new}P^T \tag{2.13}$$

If the values of the two statistics for the new data sample exceed predefined thresholds, the sample is tagged as an outlier and removed.

Of the portfolio of outlier removal methods reviewed above, we chose to use the Hampel identifier (in conjunction with the multivariate methods described ad the end of the section, as we discuss later) for data pre-treatment and outlier removal. We substantiate our choice with a relevant example in the subsection below; this example is representative of numerous validation and comparison runs carried out in the course of elaborating this thesis.

2.2.1.3 Testing on Industrial Data and Discussion

In order to compare ESD and the Hampel identifier, a pressure measurement from the industrial system was selected as the test measurement.

The pressure measurement contains high levels of noise as well as outliers from operational changes. For comparison purposes, the window size (N) and the significance level (α) were chosen to be 20 and 0.05, respectively. As shown in Figure 2.3, when short-lived outliers occur (such as brief change in production), the Hampel identifier (green) quickly detects them as outliers, but the ESD (red) identifier fails to recognize them. The upper and lower confidence intervals of the Hampel identifier do not change much by a transient set of outliers, but those of ESD identifier are greatly affected by them, as shown in the change in magnitudes of the interval. However, when an unusual event occurs and persists for a long time (Figure 2.4), both ESD and Hampel identifiers fail to detect them as outliers. Since Hampel identifier utilizes median and median absolute deviation, it is not affected by temporary outliers.

2.2.2 Filtering

2.2.2.1 Motivation

Data measurements are collected by sensors which generate electric signals, which are affected by high frequency noise. Lower frequency noise is inherent to the evolution of the process itself.

2.2.2.2 Methods

1. **Low-Pass Filter** A low-pass filter is a filter that passes frequency lower than a given cutoff frequency, but attenuates frequencies higher than the cutoff frequency [114, 97]. The cutoff frequency and the order of

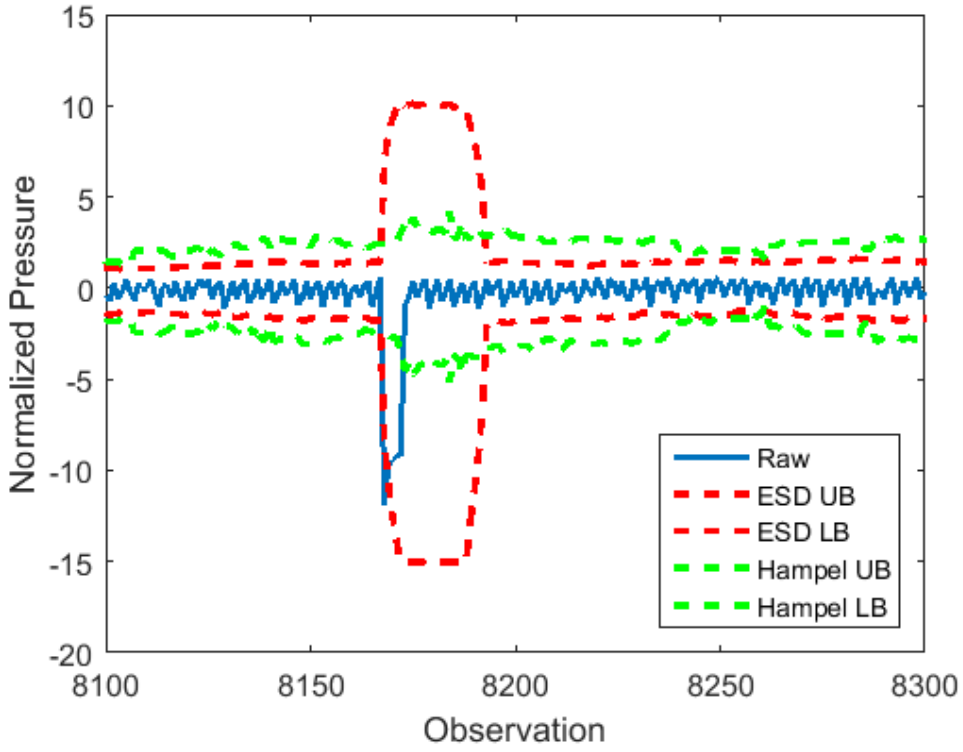

Figure 2.3: Comparison of ESD and Hampel Identifiers (Short-lived Outliers)

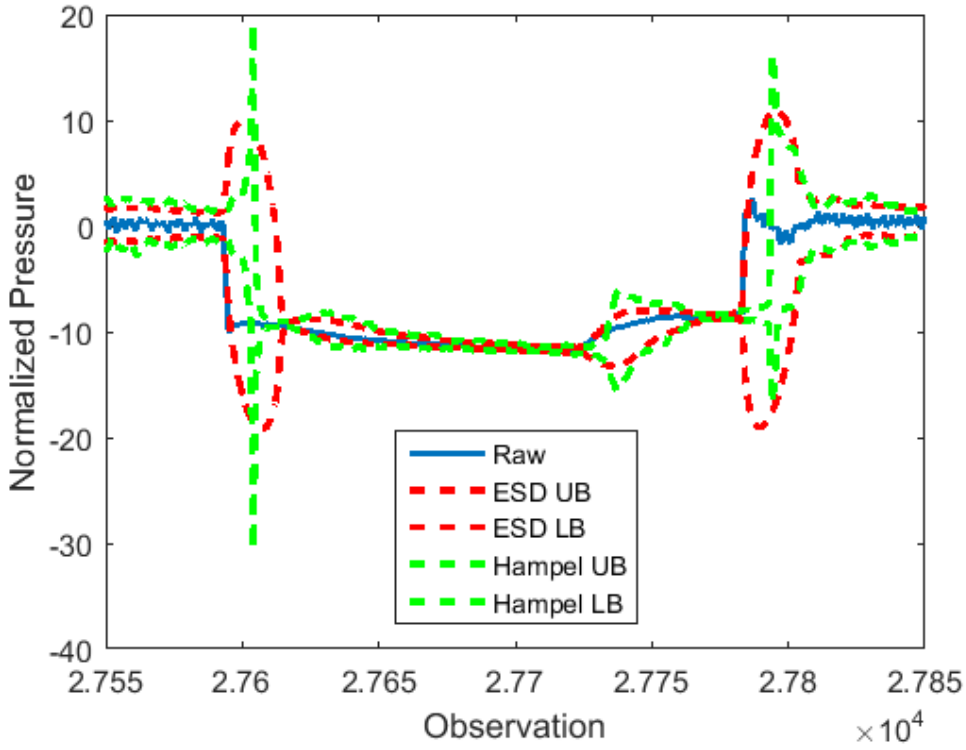

Figure 2.4: Comparison of ESD and Hampel Identifiers (Prolonged Outliers)

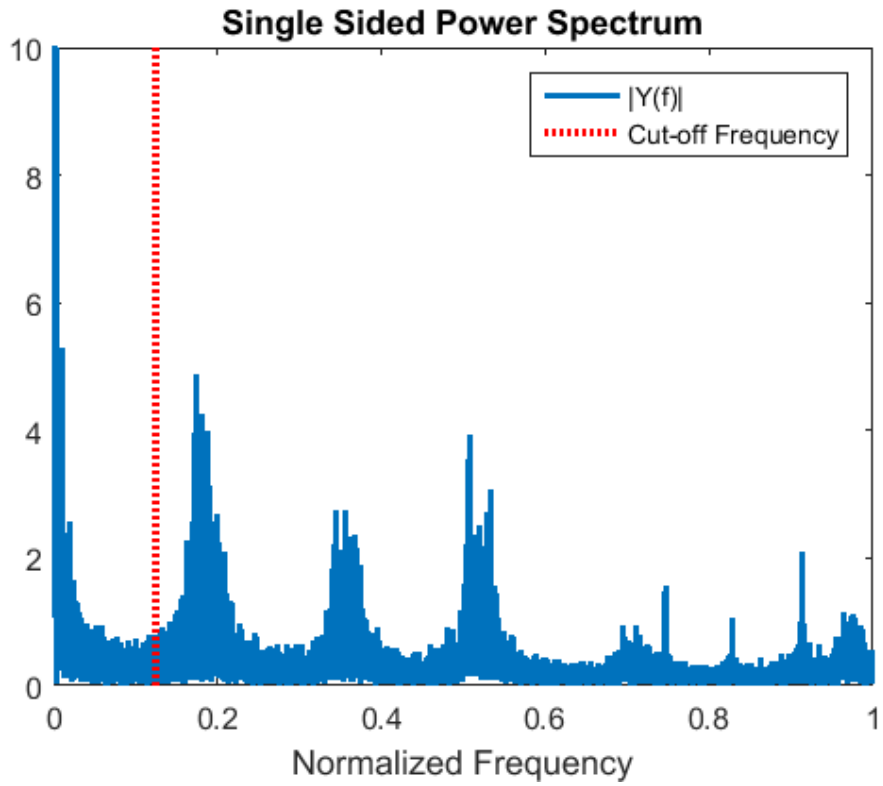

Figure 2.5: Single Sided Power Spectrum of Pressure Measurement

the filter are user defined inputs, which determine the characteristics of filters. The cutoff frequency can be determined by inspecting the power spectrum, as shown in Figure 2.5. Once the cutoff frequency is determined, the incoming data can be treated with the low-pass filter at the given cutoff frequency. The low-pass filter is a very simple, yet very powerful tool for noise removal, and has justly gained popularity in industry.

2. **Median-based Filter** The median-based filter is similar to the outlier removal methods in the previous section. Figure 2.6 shows the flow diagram of the filter. After calculating the median and standard deviation for a given window size, if a specific observation lies more than a predefined distance (typically one standard deviation) above the mean, it is replaced with the median value. This filter has two degrees of freedom that can be utilized for the smoothing effect: the window size and the threshold value. Some a priori knowledge regarding the process is required to properly assign these values. If the window size is too small, the median value might not be the actual nominal value, but might be influenced by noise; however, this can be counteracted by choosing small threshold value. Online implementation for this filter is similar to the Hampel identifier. Once the statistical parameters such as μ and standard deviation are determined, the filter can be in place with the moving window.

3. **Savitzky-Golay Filter** Lastly, the Savitzky-Golay (S-G) filter smooths the signal by using convolution with sub-sets of adjacent data points [112]. Rather than defining the properties in Fourier domain and then translating back to time domain (low-pass filter), the S-G filter operates strictly in the time domain. The filter works as follows:

$$g_i = \sum_{n=-n_L}^{n_R} c_n f_{i+n} \tag{2.14}$$

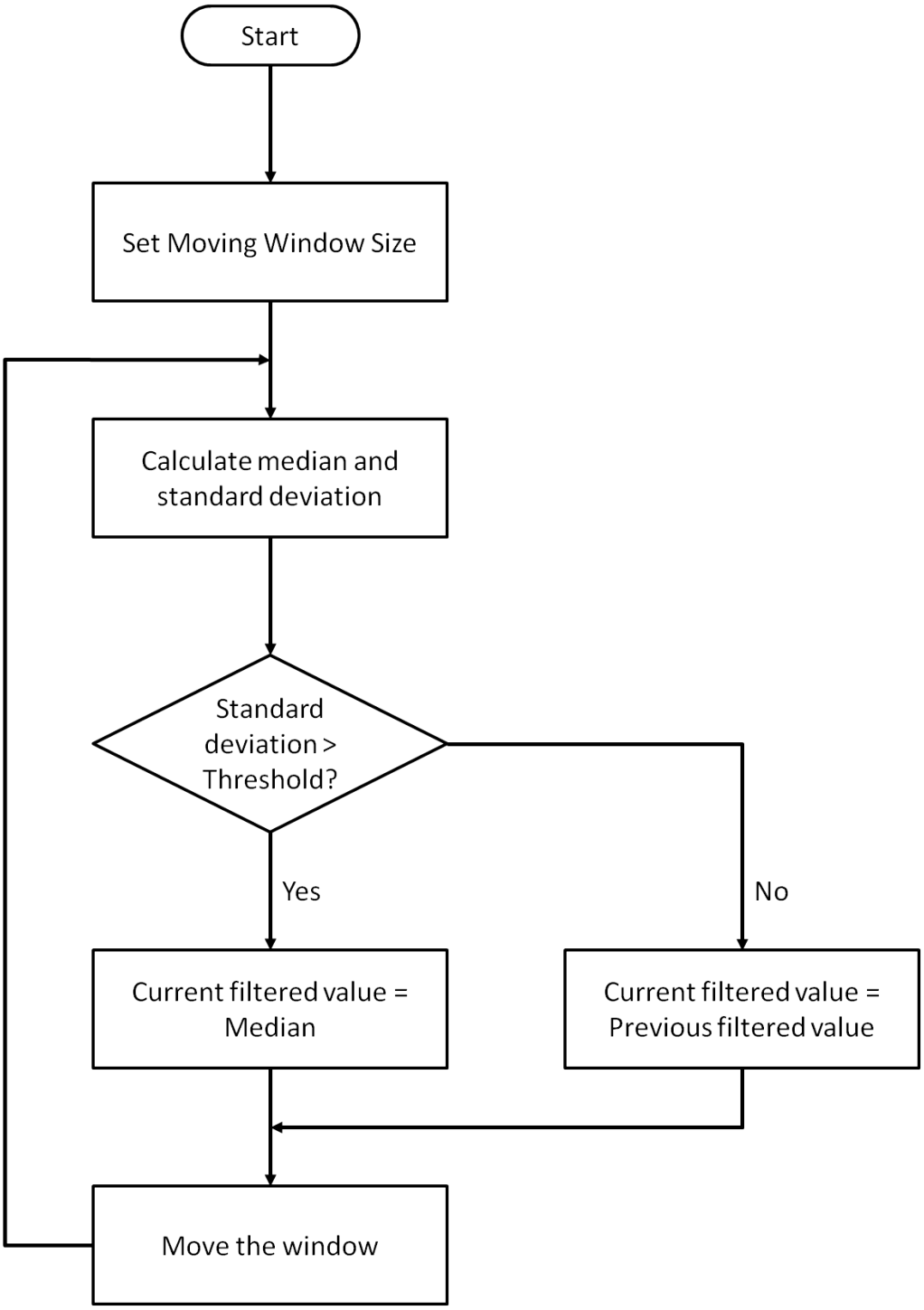

Figure 2.6: Flow Diagram of Median-based Filter

The observed value f_i is replaced by a linear combination of g_i of itself and the points nearby it. n_L and n_R determine the number of points used to the "left" and "right" of the data, respectively. The idea behind S-G filter is to determine the filter coefficients, c_n, such that the tabulated coefficients yield a good fit. Instead of using constant values, which is just finding average, the coefficients are determined by a polynomial of higher order. Using the given data points, least-squares fit of a higher order polynomial is performed, and the g_i is found using the value of the polynomial at that position. For the S-G filter, the order of the polynomials, which determines c_n, and the window size (n_L and n_R) are tuning parameters that can determine how the filter behaves. The choice of polynomial order and number of coefficients need to be balanced in order to reduce noise, but not to distort the data. Also, these tuning parameters need to be assigned with some care. The L data points are used to approximate c_n for Nth order polynomial function, which means that L points are used to compute $N + 1$ coefficients. If $N + 1 = L$, there is no filtering and smoothing effect resulting from S-G filter. Generally, N has to be much smaller than L in order to achieve filtering and numerical stability. If $N << L$, the filter is much more aggressive and more smoothing will take place. Unlike the previous cases, the online implementation of this filter is not as simple as using the predefined parameters. Because Savitzky-Golay filter incorporates "future" data points, in order to implement online, n_R can be set to 0, which then the

filter will only depend on the "past" measurements.

2.2.2.3 Testing on Industrial Data and Discussion

Similar to the outlier removing techniques, the filtering methods were compared using the pressure datum. The cutoff frequency was determined as 12.5% of its maximum frequency (as shown in Figure 2.5). And 6th order Butterworth filter was used. For the median-based filter, a window size of 200 and a threshold value of 30 were used. For the S-G filter, n_L and n_R were set as 100, and a 2nd order polynomial was used. All three filtering techniques successfully remove high frequency noise and are able to produce smoothing effect similarly, as shown in Figure 2.7. However, the low-pass filter and median based filter introduce time delays, as shown in Figure 2.8. Also, the median-based filter is computationally expensive, as all the values of the moving windows have to be sorted to find the median value. On the other hand, the S-G filter does not introduce any time delay, and is not computationally cumbersome because the coefficients can be found simply by solving –offline– a set of linear equations.

Based on these studies, we decide to use the S-G filter for processing all the data used in the analyses presented in the sequel in this thesis.

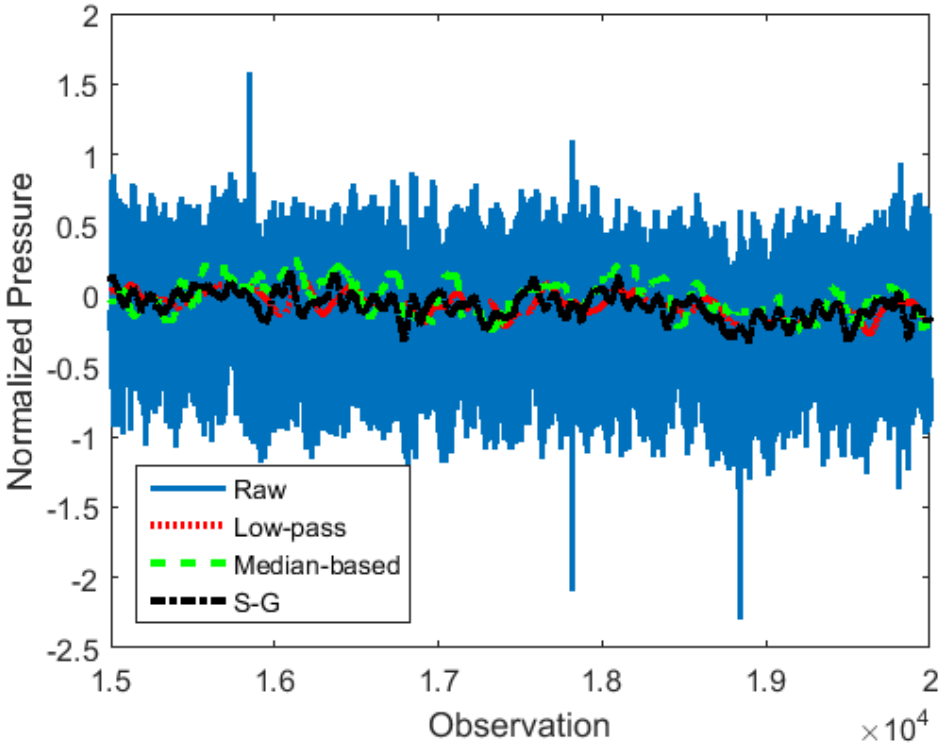

Figure 2.7: Comparison of Low-pass Filter, Median-based Filter, and Savitzky-Golay Filter

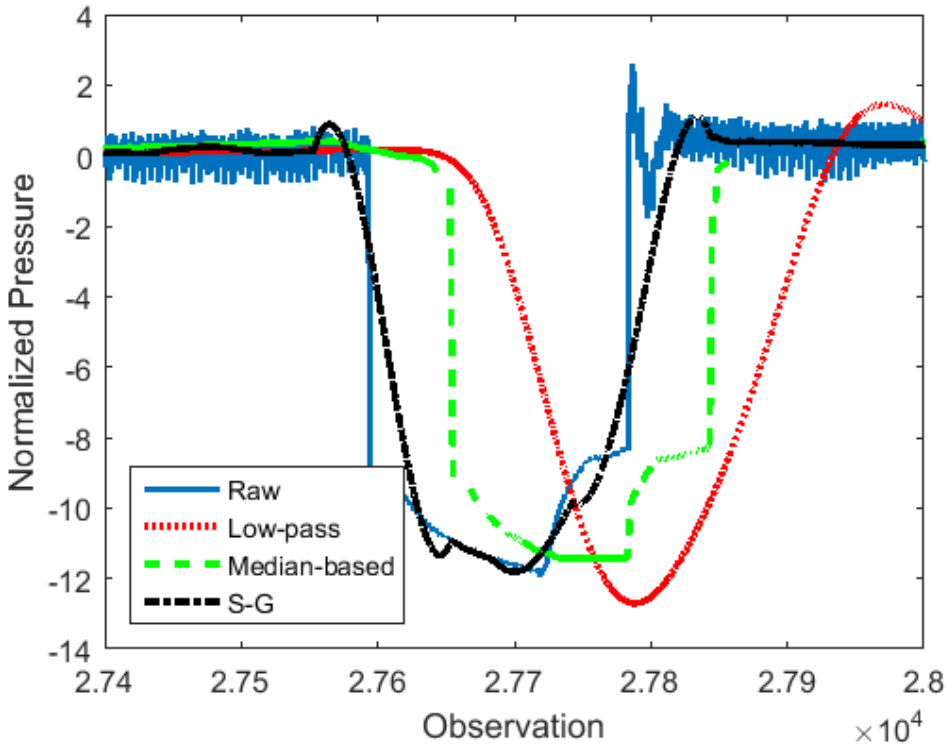

Figure 2.8: Comparison of Low-pass Filter, Median-based Filter, and Savitzky-Golay Filter (When Change Occurs)

2.3 Aligning Batch Data to Continuous Data

2.3.1 Motivation

An ongoing research thrust in complex chemical processes concerns the development and application of data driven regression methods to reduce the dimensionality of the data, while trying to capture the process behavior. Techniques such as principal component analysis (PCA) have been applied and used widely, along with specific modifications [59]. Data-driven modeling methods for batch processes [96, 85, 94, 95] were developed based on previous approaches for continuous systems [59, 60, 106, 105, 109, 72]; however, the two sub-areas have since evolved quite separately. In a different vein, it is worth mentioning that PCA and PLS have been adapted to address issues such as dynamics and nonlinearity [113, 132, 115, 110, 78, 77].

However, there has been very little emphasis placed thus far on data-driven modeling of B2C processes in spite of their relatively widespread use in industry. Here, we recall the work of Choulak et al., who developed a dynamic model of a plastic processing system with reaction by considering a single spatial dimension and modeling each barrel (or section) of the plug flow reactor as a separate continuously stirred tank reactor (CSTR) [18]. The authors take advantage of temperature measurements in different barrels to facilitate parameter estimation [18]. Bahroun et al. utilized a similar modeling technique to describe a three-phase catalytic slurry intensified continuous chemical reactor, and developed control strategy for the system. They devise the optimization and control as a hierarchical control structure where the

optimization layer computes the setpoints for the controller [6]. These developments, however, do not consider the batch and continuous sections of the process in an *integrated* fashion; rather, they focus on the batch and continuous section separately [6, 18].

As we have emphasized above, one of the main difficulties in analyzing B2C processes comes from the data gaps between the batches, and the need to align these data to the (uninterrupted) readings obtained from the continuous section. In order to align the batch data to the continuous data, the gaps between the batch data need to be filled.

2.3.2 Method

Batch data has three dimensional data structure (variable, time, and batch). This data structure can be unfolded into a two dimensional representation. Batch-wise unfolding consists of preserving the trajectory of each variable within all the batches, and is typically carried out in conjunction with time warping - the latter comprising a calculation aimed at representing the time evolution of the variables over a common, unified time horizon. Batch-wise unfolding (Figure 2.9) has proven to be a popular approach in the PCA research community, as it supports analyses of batch process performance and fault detection [94, 127, 131].

However, our interest in this work lies in correlating the batch and continuous sections of a B2C process and, as such, we are not concerned with preserving the batch nature of the data. Rather, we desire to create

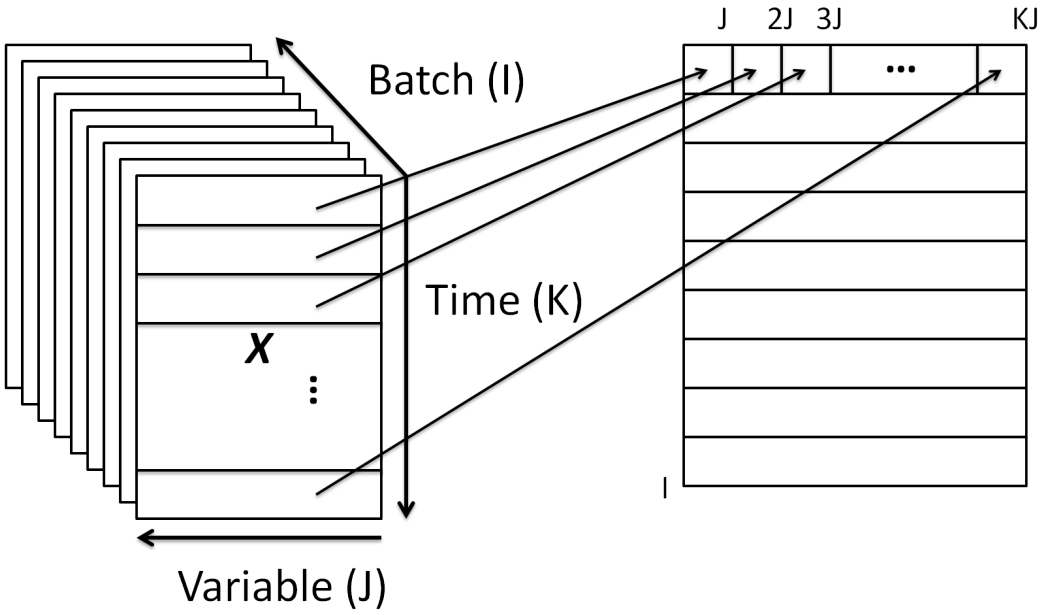

Figure 2.9: Batch-wise Unfolding

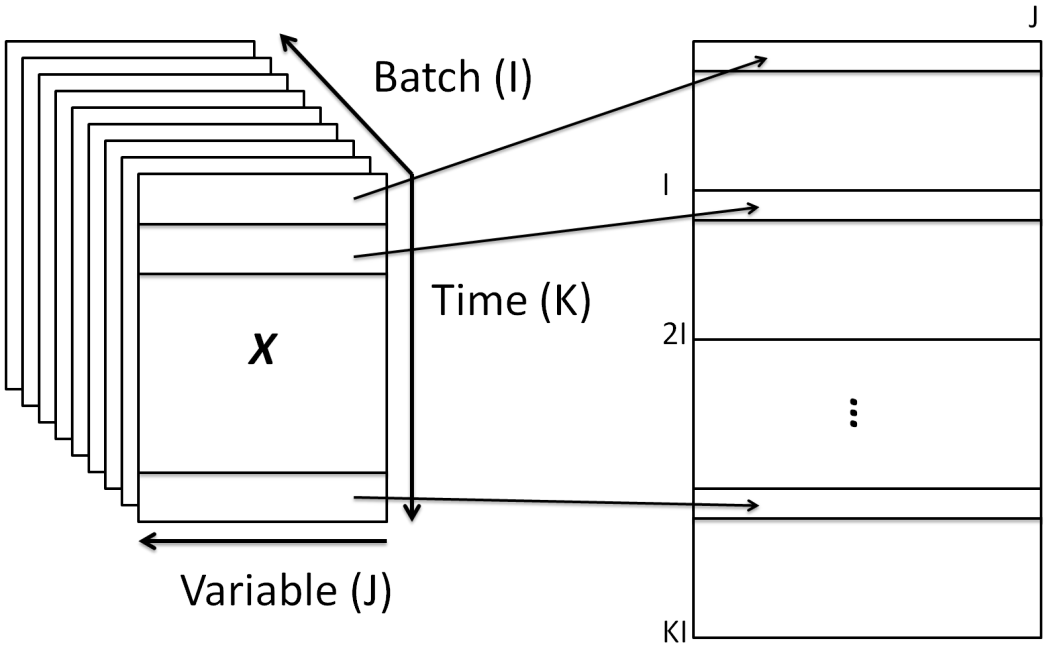

Figure 2.10: Variable-wise Unfolding

a continuous-like representation of the batch section data, which call for a variable-wise unfolding of the respective data sets (Figure 2.10).

This unfolding exposes the gaps between the batches, which need to be treated. One simple, yet flawed solution is to simply remove the observations between the gaps in batches and the corresponding measurements in continuous data set; however, this reduces the number of observations, which is undesirable for data analysis, and also leads to information loss. Another solution is to just use the data without modification, with the potential pitfalls of, e.g., the "gap" periods being considered as process faults or, alternatively, resulting in a model with a very high tolerance/threshold for process

faults and failures (note that the data recorded during these "gaps" are typically not blank database entries; rather, a constant value (such as -1 or 0) is recorded).Our approach to filling these gaps is based on a physical observation, namely, that once a batch has been completely fed to the continuous section (but the subsequent batch is not yet being processed), the values of the process variables from the *batch* section no longer change as far as the continuous process is concerned. Thus, it is natural to fill the data gaps with "characteristic" (with this notion defined more rigorously below) values of variables, as determined from the *most recent batch being processed in the continuous section.*

In this manner, batch to batch variability is maintained, the sample count of the batch measurements matches that of continuous counterpart, and the information needed to correlate the batch to continuous data is not compromised.

This batch characteristic values can be chosen differently in order to suit different situations. For example, in case of batch reaction followed by continuous process, the end concentration and final temperature can be the batch characteristic variables. Other examples include the particle size or molecular weight distribution at the end of the batch, assuming that measurements of these variables are available.

On the other hand, we noted above that such measurements of the batch output may not be available. In this case, trajectories or time-average values of the *inputs* to the batch process may be used as the (characteristic)

data to represent the batch section of the B2C process.

2.3.3 Testing on Industrial Data and Discussion

In order to demonstrate the method described above, the input property measurements of the batch section of the industrial system were utilized. The current setup lacks the quality variable measurement that accurately portrays the condition of the batch output. Instead, a measurement of the feed of the batch system in the form of a material property distribution is collected. Figure 2.11 shows a histogram of this distribution, and Figure 2.12 illustrates the evolution in time of each discrete bin measurement.

In the given example of batch distribution, the average values of all the bin measurements over steady state region of the batch were calculated and used as the batch characteristic value to fill the data gaps. The batch operation in this process tries to maintain the distribution to be consistent thus the average values were used to reduce the measurement noise. Figure 2.13 shows only two variables of the distribution to facilitate the visualization.

2.4 Summary

In this chapter, some data preprocessing techniques, which include data cleaning, outlier removal, filtering, and batch data alignment to continuous data are discussed. These preprocessing steps need to be carried out carefully as a precursor to the data analysis steps described later in the thesis.

We contributed a method for aligning the batch data to continuous

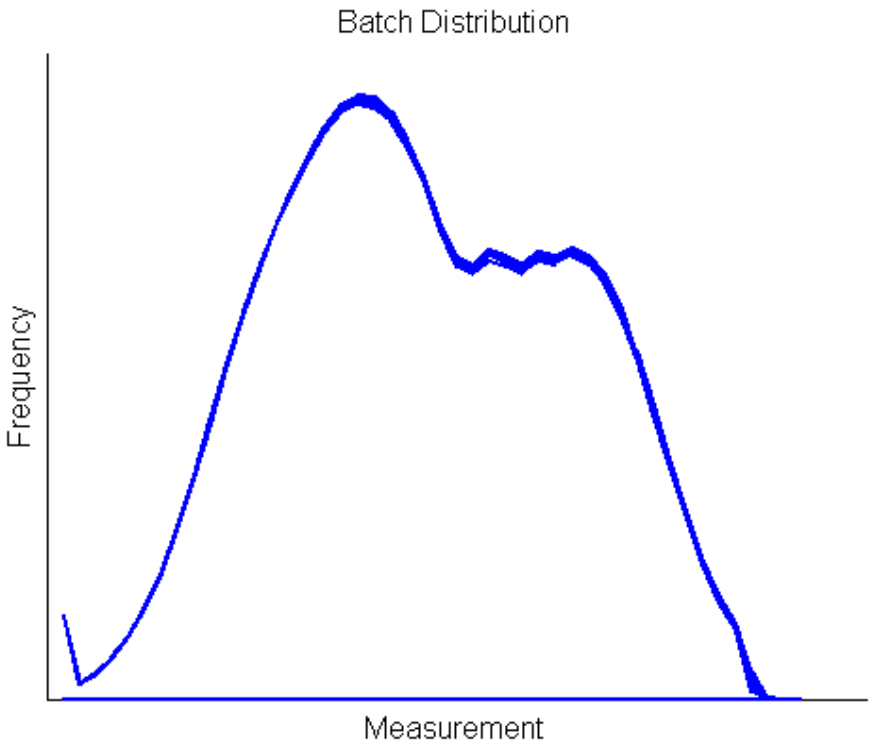

Figure 2.11: Distribution of Batch Quality Measurements

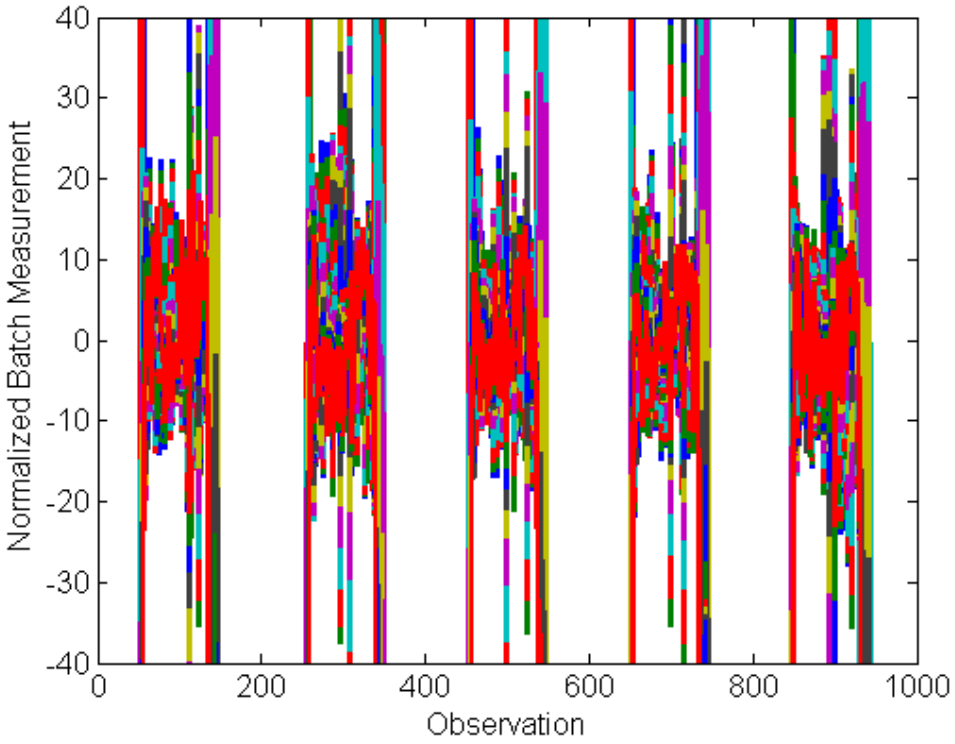

Figure 2.12: Variable-wise Unfolded Batch Distribution Measurement

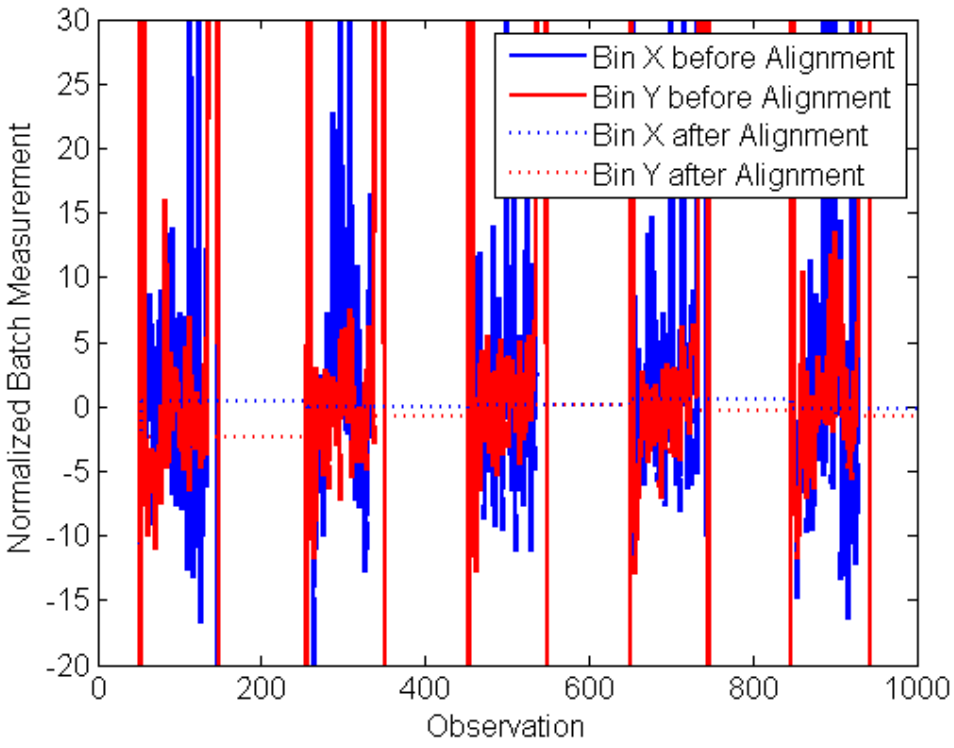

Figure 2.13: Aligning Variable-wise Unfolded Batch Measurement by Filling in the Gaps with Batch Characteristic Variables

data. In particular, variable-wise unfolded batch data are used and a characteristic value of each variable is used to represent the batch in the "gap" periods. This value acts as a representative value of an entire batch from the perspective of the continuous process, and is used until the following batch starts to get processed. This approach captures batch to batch variability, and the impact of the batch subsystem on the downstream continuous process.

Chapter 3

Data-Driven Modeling and Variable Selection

3.1 Introduction

With the recent advancement in computer processing power and data storage, it has become easier and cheaper to store and process data. Gordon E. Moore observed that the number of transistors in a circuit doubles every two years, and predicted that this trend will follow in the future [93]. This trend has led to not only cheap and numerous sensors, which have facilitated data measurement and storage, but also faster computing power and efficiency, which have enabled complex algorithms and faster calculations. In chemical engineering, abundant data have allowed for better understanding in process and improvement in process operation and safety.

With the plethora of data, data-driven techniques have been utilized to model and estimate variables and process conditions that are difficult or expensive to measure. These techniques are referred to as "soft sensors" and can estimate and predict the current status of the process, and also monitor and diagnose the operating condition of the process. These methods have been shown to be useful in many industries [105, 42, 84, 63]. On-going research in complex chemical processes utilize data driven regression methods to reduce

the dimensionality of the data.

In this chapter, we first focus on data-driven modeling of B2C processes using tools such as PCA and PLS. At first, we use these tools to carry out performance evaluation for the batch and continuous systems individually. Thereafter, we develop analysis tools for the entire B2C process based on the aligned data obtained using the techniques described in the previous chapter. We also incorporate a clustering algorithm to account for the multiple operating modes of the process.

In the second part of the chapter, we consider variable selection techniques for the latent variable-based PCA/PLS algorithms. We compare and contrast techniques that fall into the filter and wrapper methods category, including variable importance in projection (VIP) filtering, beta coefficient filtering, uninformative variable elimination (UVE), stepwise elimination, and Monte Carlo uninformative variable elimination (MCUVE). These techniques not only reduce the number of variables that need to be retained for modeling purpose, but also improve the accuracy of the model.

3.2 Data-Driven Modeling

3.2.1 Motivation

Due to the complexity of the sequential batch-continuous process, it is difficult to develop a first-principles model to describe the entire system accurately (see also discussion in Section 2.1). Instead, data-driven modeling techniques are used. These methods are utilized for process monitoring and

fault detection [28, 67, 86]. The most prevalent and applicable examples of the data-driven modeling methods are PCA and PLS. There have been many other various types that originate from these two methods for specific applications. Most of these methods focus on batch and continuous processes separately; however, in sequential batch-continuous process, both the batch and the continuous portions affect on the final product quality. Therefore, in this study a new method to find the correlation between the batch and continuous parts of the process is explored.

In the chemical industries, numerous types of data-driven modeling techniques are used to capture different types (batch and continuous) and different characteristics (dynamics and steady-state) of process. Lee et al. used kernel principal component analysis (KPCA) developed by Scholkopf et al. to model and monitor a wastewater treatment process, which is highly nonlinear [77, 113]. Nomikos et al. developed multiway principal component analysis (MPCA), which unfolds the three dimensional batch data to two dimensions in order to perform PCA on the data matrix [94, 95]. In order to incorporate dynamic effects into modeling, Ku et al. developed dynamic principal component analysis (DPCA), which utilizes time lag shift method to PCA [72]. Zhang et al. developed a MPCA that incorporates dynamics of the batch by utilizing exponentially-weighted moving average (EWMA) [133]. Many more techniques, similar to the ones mentioned above, have been developed to supplement and improve the existing methods [23, 24, 57, 76, 75, 109, 127, 131, 132].

3.2.2 Methods

In the previous chapter, we briefly introduced how PCA is utilized for outlier removal and process monitoring using Hotelling's T^2 and Q-statistics. In this section, we review the method in more detail. Also, partial least squares (PLS) regression, which exploits the projection of both input and output variables in a latent-variable space, is introduced.

1. **Principal Component Analysis (PCA)**

   As mentioned in Section 2.2.1.2, PCA is a linear variable transformation, which projects a large number of variables that might be correlated into a new coordinate system, obtaining a set of linearly independent principal components (PCs). In order to reap the dimensionality reduction benefit of this transformation, the number of PCs retained can be chosen to be lower than the number of variables in the original data set. This is done by ordering the components in decreasing order of the amount of variance captured. In typical high-dimensional data scenarios, the first few components capture a significant (often disproportionately large) amount of the total variance of the data set; as a consequence, the dimensionality of data can be reduced considerably without a significant loss of information. The resulting coordinate transformation matrix (the "PCA loadings") can be used to project any new data samples, and the statistical information concerning the fit of the model (Hotelling's T^2 statistic) and the modeling error (the Q-statistic) are available for pro-

cess performance evaluation and fault detection purposes [59].

While extremely appealing from this point of view, the performance of such projection methods is vitally dependent on the data that are used in constructing the initial model. Typically, data from a "golden period"/reference of "normal" operation are employed [106]. However, this reference data set is often chosen subjectively based on operator experience and opinions, and may or may not be fully representative of the process operations. More importantly, "normal" operation from the operator perspective does not necessarily mean economically optimal operation and. As a consequence, subsequent analysis and control decisions made based on such a model may result in perpetual or long term operation that remains economically mediocre.

Prior to applying PCA decomposition, the original data set $\mathbf{X}$ is subjected to two standard preprocessing steps, mean centering and unit variance scaling. Mean centering re-centers each variable to a mean value of 0 by simply subtracting the mean value from all the observation. Unit variance scaling entails dividing all the observations by their respective standard deviations. Mean centering and unit variance scaling result in all scaled variables with 0 mean value and 1 standard deviation, and can be applied online to new data samples using the mean and standard deviation values obtained from the reference data set.

Mean centering and unit variance scaling are collectively referred to as "auto-scaling."

After auto-scaling, PCA can be carried out. Let $\mathbf{x} \in \mathbb{R}^m$ be a set of observations of m variables. Now, assume that the data is collected for N number of observations, which will result in $\mathbf{X} \in \mathbb{R}^{N \times m}$. PCA decomposes $\mathbf{X}$ to a score matrix $\mathbf{T}$ and a loading matrix $\mathbf{P}$, as shown in Equation 3.1.

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E} \tag{3.1}$$

The covariance matrix can be found using the following equation:

$$\mathbf{S} = \frac{1}{N-1}\mathbf{X}^T\mathbf{X} = \mathbf{P}\Lambda\mathbf{P}^T \tag{3.2}$$

and $\Lambda = \frac{1}{N-1}\mathbf{T}^T\mathbf{T} = diag\{\lambda_1, \lambda_2, ..., \lambda_m\}$, where Λ is the diagonal matrix of the eigenvalues in the PCA decomposition. The number of observations in PCs is equal to the number of observations in original matrix.

Dimensionality reduction is achieved by retaining a smaller number ($p < m$) PCs that capture high variance in the data. To this end, the Λ matrix is sorted in descending order. The percent of variance retained as a function of the number of components p retained is determined as:

$$\%\text{ variance retained}(k) = \frac{\sum_{i=1}^{p} \lambda_i}{\sum_{i=1}^{m} \lambda_i} \tag{3.3}$$

Different heuristics are used to determine the number of PCs retained p. The simplest approach is to use a constant value such, e.g., 0.9, in which

case enough components are retained to capture 90% of the variance of the original data set. A more rigorous method involves cross-validation, an iterative procedure that involves, i) building a PCA model using only a subset of the available N observations in the reference data set, and, ii) using the remaining data to check the model prediction (i.e., performing model validation).

The Q- and Hotelling's T^2 statistics reviewed in the previous chapter are used to perform cross-validation. A large value of the Q-statistic means that the current PCA model is unable to explain the large variance in the considered data point. Conversely, a data sample is considered normal if the following condition holds.

$$Q_i \leq Q_\alpha \tag{3.4}$$

Q_α is the control limit at significance level α. Q_α is defined as follows [60]:

$$Q_\alpha = \theta_1 \left[\frac{c_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right]^{1/h_0} \tag{3.5}$$

c_α is the normal deviate corresponding to the upper $(1 - \alpha)$ percentile. θ_i and h_0 are defined as follows:

$$\theta_i = \sum_{j=p+1}^{m} \lambda_j^i, \qquad i = 1, 2, 3 \tag{3.6}$$

$$h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2} \tag{3.7}$$

where p is the number of principal components retained and λ_j corresponds to the eigenvalue of component j in the PCA.

In order to define Hotelling's T^2 limits, the following assumptions have to be made: the data are normal and follow the multivariate Gaussian distribution. Then, the T^2 statistic follows a F-distribution of the following form:

$$\frac{N(N-A)}{A(N^2-1)}T^2 \sim F_{\alpha,A,N-A} \tag{3.8}$$

where $F_{\alpha,A,N-A}$ is a F-distribution with $(A, N-A)$ degrees of freedom, N is the number of samples, and A is the number of principal components. Using this the control limit T_α^2 can be defined.

$$T^2 \leq T_\alpha^2 = \frac{A(N^2-1)}{N(N-A)}F_{A,N-A,\alpha} \tag{3.9}$$

2. **Partial Least Squares Regression (PLS)**

Partial least squares (PLS) regression relies on the same principles as PCA, aiming to establish a (linear) input-output relationship between a predictor vector $\mathbf{X}$ and $\mathbf{Y}$, a vector of response/output variables. A review of PLS approaches is presented in Andersson et al. [5], while specific chemical industry applications are discussed by Wold et al. and

Nomikos et al. [95, 127]. Fundamentally, PLS consists of projecting both the predictor and response variables, and establishing a linear relationship between (a subset of) the projected variables P and Q.

$$\begin{aligned} \mathbf{X} &= \mathbf{T}\mathbf{P}^T + E \\ \mathbf{Y} &= \mathbf{U}\mathbf{Q}^T + F \end{aligned} \tag{3.10}$$

where $\mathbf{X} \in \mathbb{R}^{m \times n}$ is the input, $\mathbf{Y} \in \mathbb{R}^{m \times r}$ is the output, $\mathbf{T} \in \mathbb{R}^{m \times p}$ and $\mathbf{U} \in \mathbb{R}^{m \times p}$ are the score matrices, and $\mathbf{P} \in \mathbb{R}^{n \times p}$ and $\mathbf{Q} \in \mathbb{R}^{n \times p}$ are the loading matrices. Similar to PCA, each score vector, $\mathbf{t}_i$, is a linear combination of $\mathbf{X}$ (however, not necessarily orthogonal) and $\mathbf{u}_i$ is $\mathbf{Y}$-score vector that is linearly dependent on $\mathbf{T}$. The number of components in PLS can be determined similarly to PCA. Many techniques such as using cross-validation or the Akaike Information Criterion (AIC) are discussed by Kramer et al. [68].

$\mathbf{Y}$ can be calculated linearly using $\mathbf{X}$ and $\tilde{\beta}$ by using $\mathbf{Y} = \mathbf{X}\tilde{\beta}$. Slight discrepancies between PLS algorithms are due to the different ways of estimating the factor and loading matrices $\mathbf{T}, \mathbf{U}, \mathbf{P}, \mathbf{Q}$, which affect $\tilde{\beta}$. $\tilde{\beta}$ can be calculated as follows:

$$\tilde{\beta} = \mathbf{R}(\mathbf{T}^T\mathbf{Y}) = \mathbf{R}\mathbf{R}^T\mathbf{X}\mathbf{Y} \tag{3.11}$$

where $\mathbf{R}$ is the weight matrix, which differs in different PLS algorithms. Similar multivariate statistics from PCA can be applied to PLS as well.

$$T^2 = \mathbf{t_0}^T \mathbf{\Lambda}^{-1} \mathbf{t_0} \sim \frac{A(n^2-1)}{n(n-A)} F_{A,n-A}$$
$$Q = \|\mathbf{x_0} - \mathbf{t_0}\mathbf{p_0}\|^2 \sim g\chi_h^2 \tag{3.12}$$

where $\mathbf{t_0}$ and $\mathbf{p_0}$ are scores and loadings for PLS, respectively, A is the number of principal components retained for the PLS model, and $\Lambda = \frac{1}{n-1}\mathbf{T}^T\mathbf{T}$. Given the significance level α, $F_{A,n-A}$ and χ^2 can be obtained from the Fischer and the χ^2 distributions for T^2 and Q statistics. Similar to PCA, Q-statistic measures the model residual of the data, and the Hotelling's T^2 statistic measures the mean shifts within the score vectors.

3. **Kernel PCA/PLS** PCA and PLS are linear analysis methods. While they can handle mild nonlinearities associated with the underlying phenomena described by the data, dedicated "kernel" reformulations of these approaches are indicated to deal with systems exhibiting nonlinear behavior. One of the best known such modifications is Kernel PCA (KPCA) developed by Schölkopf et al [113]. By using integral operator kernel functions, PCA is carried out in high-dimensional feature spaces, related to the input by a nonlinear mapping. The method is as follows, inputs are mapped into a dot product space $\mathbb{F}$ called feature space, and then PCA is performed. Using the covariance matrix in $\mathbb{F}$,

$$\bar{C} = \frac{1}{M}\sum_{j=1}^{M} \Phi(\mathbf{x_j})\Phi(\mathbf{x_j})^T \tag{3.13}$$

where $\Phi(\cdot)$ is a nonlinear kernel function that transforms $\mathbf{x}$ into $\mathbb{F}$.

Utilizing Equation 3.13, eigenvectors in the feature space can be obtained by the following equation.

$$\lambda \mathbf{V} = \bar{C}\mathbf{V} = \frac{1}{M}\sum_{j=1}^{M} \Phi(\mathbf{x_j})\mathbf{V}\Phi(\mathbf{x_j})^T \tag{3.14}$$

The solutions $\mathbf{V}$ lie in the span of $\Phi(\mathbf{x}_1), \ldots, \Phi(\mathbf{x}_M)$. Then, there exist $\alpha_i \quad (i = 1, \ldots, M)$ such that

$$\mathbf{V} = \sum_{i=1}^{M} \alpha_i \Phi(\mathbf{x}_i) \tag{3.15}$$

Kernel representations are of the form:

$$K_{ij} = (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)) = k(\mathbf{x}_i, \mathbf{x}_j) \tag{3.16}$$

By multiplying $\Phi(\mathbf{x}_k)$ on both sides of Equation 3.14, we get

$$\begin{aligned} &\lambda \sum_{i=1}^{M} \alpha_i (\Phi(\mathbf{x}_k) \cdot \Phi(\mathbf{x}_i)) = \\ &\frac{1}{M}\sum_{i=1}^{M} \alpha_i (\Phi(\mathbf{x}_k) \cdot \sum_{j=1}^{M} \Phi(\mathbf{x}_j))(\Phi(\mathbf{x}_j) \cdot \Phi(\mathbf{x}_i)), \qquad k = 1, \ldots, M \end{aligned} \tag{3.17}$$

And using Equation 3.16,

$$M\lambda\alpha = K\alpha \tag{3.18}$$

Table 3.1: Kernel Functions used in KPCA/KPLS

Kernel Functions	Function Form
Polynomial	$k(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^d$
Radial Basis	$k(\mathbf{x}, \mathbf{y}) = \exp(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2})$
Neural Network	$k(\mathbf{x}, \mathbf{y}) = \tanh((\mathbf{x} \cdot \mathbf{y}) + b)$

This allows the computation of the dot product in $\mathbb{F}$ without having to carry out the map Φ. Table 3 shows the different kernel functions that can be used for mapping into the feature space.

KPLS is similar to KPCA, in the sense that $\mathbf{Y}$ is also mapped into a nonlinear feature space, and then PLS is applied. The main challenge of using KPLS or KPCA is the proper choice of the nonlinear kernel. Ideally, this function is selected based on some knowledge of the underlying physical phenomena of the system under consideration; lacking such knowledge, the selection of the kernel function amounts to a trial-and-error effort [5]. Also, when mapped into the feature space, the dimensionality of the input can be infinite, which might not result in dimensionality reduction, which is the benefit of using PCA and PLS [113].

4. **Multiway PCA/PLS**

Multiway PCA/PLS is a special type of numerical analysis tool that handles 3-dimensional batch data [94, 127, 131]. Batch data are unique in that they can be regarded as a three dimensional data structure with variable, time, and batch. Data are unfolded batch-wise or variable-wise

as was discussed earlier, as shown in Figures 2.9 and 2.10.

After unfolding the 3-dimensional data cube into a 2-dimensional data matrix, PCA or PLS can be carried out for modeling and monitoring of the batch process. In this analysis, the residual is the deviation of the batch trajectory from the average batch trajectories that were used for training the model. Batch-wise unfolding can naturally handle nonlinear batch trajectory; however, in order to perform batch-wise unfolding and carry out MPCA/MPLS, time warping has to be performed (see page 33). This is disadvantageous for online implementations, as one has to estimate or predict the future values of the batch [94, 96]. There are other methods, such as hybrid-wise unfolding, which unfolds the data batch-wise, preprocesses the data, and rearranges the data to the variable-wise structure to reap benefits of both batch-wise and variable-wise unfolding [75, 133].

There are other numerous variants of PCA/PLS techniques such as Dynamic PCA (DPCA) [110, 78] and Dynamic Batch PCA (DBPCA) [132]. They all have different preprocessing techniques or rearranging prior to performing PCA/PLS in order to capture different phenomena (i.e. dynamics); however, they all utilize the same PCA and PLS methods reviewed in this section.

3.2.3 Local Batch Monitoring

In sequential batch-continuous process, the batch side of the process can be monitored individually/locally. The goal is to monitor and detect any changes or disturbances that occur in the batch process. In the industrial system considered in this thesis, the distributional properties of the raw material (e.g., molecular size distribution, particle size distribution) are monitored. Figure 3.1 represents the distributional property of the raw materials that are fed to the batch process. In this setup, the distribution is measured by more than 40 discrete bin measurements. In Figure 3.1, each line represents a distribution measurement of the raw material. Each batch has very similar distributional profiles with slight variations in two peaks. The figure shows data from multiple campaigns, each comprising multiple batches. Note that there are slight differences between data collected in different campaigns as well: the blue lines represent batches from one campaign and the red lines represent batches from another campaign. The changes in distribution shape shown in Figure 3.1 can be attributed, e.g., to ambient disturbances (e.g., changes in humidity and temperature) or to changes in the raw materials themselves. If this data set is variable-wise unfolded and plotted, we obtain Figure 3.2.

As discussed in the previous chapter, the gaps in the batch data are filled and PCA is performed. After PCA is performed, two principal components are retained. Figure 3.3 shows that using two principal components out of 40+ input variables, about 80% of the variability within the data set can be captured. Also, by examining the loadings of these principal components,

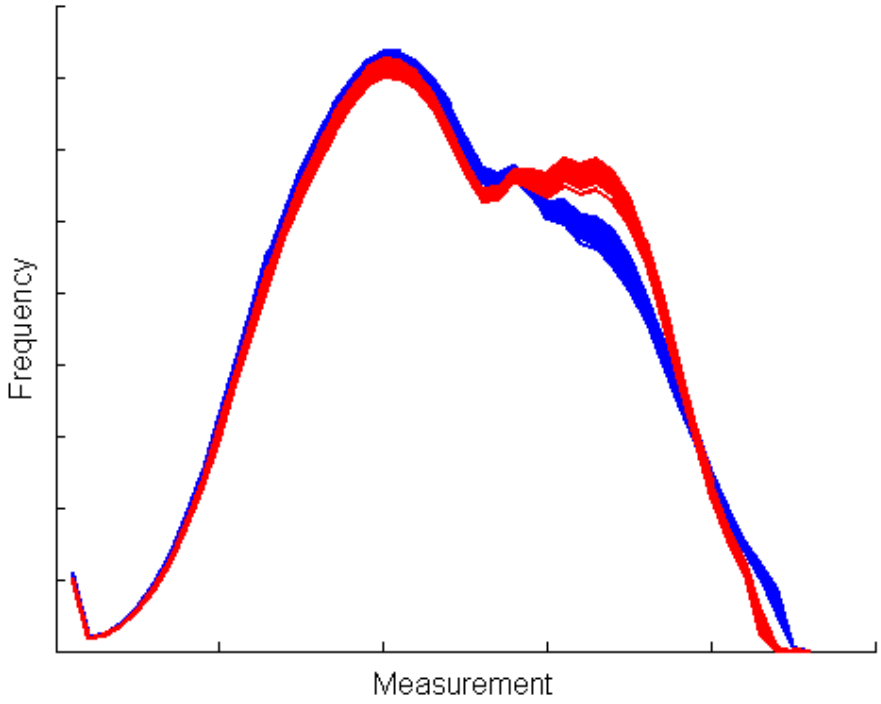

Figure 3.1: Sample Distribution of Batches on Two Different Production Runs

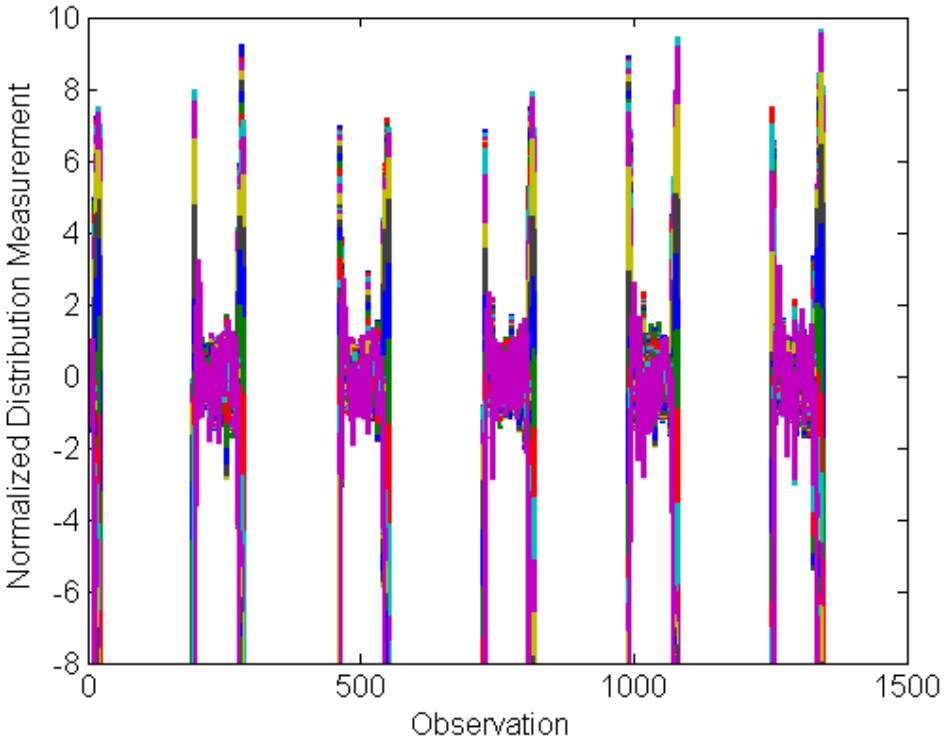

Figure 3.2: Variable-wise Unfolded Batch Distribution

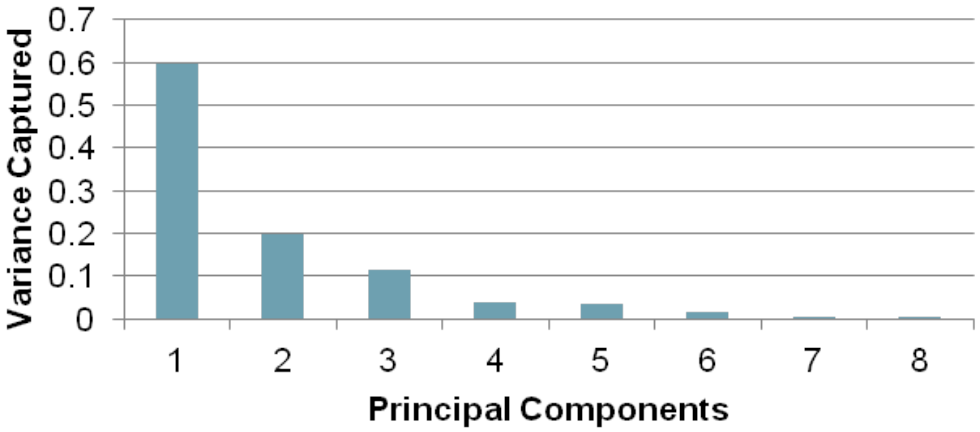

Figure 3.3: % Variance Captured with Principal Components

more insights can be obtained. Figure 3.4 shows the loadings of principal components 1 and 2. These two components represent how the shape and the skewness of the distribution change. The loading of the first principal component indicates that if the score of this principal component is high, the distribution is much more uniform and flat. Similarly the second principal component indicates the shift in distribution. If the score of this component is high, the distribution will shift to the left. This facilitates the monitoring of the raw material properties.

After PCA is performed and the number of PC retained is determined, the Hotelling's T^2 and Q-statistics can be used to diagnose the current status of the batch distributions. Figure 3.5 shows the Hotelling's T^2 and the Q-statistics control charts. The "abnormal" batches are are colored red to show how they are identified in these two statistical measures. Both Hotelling's

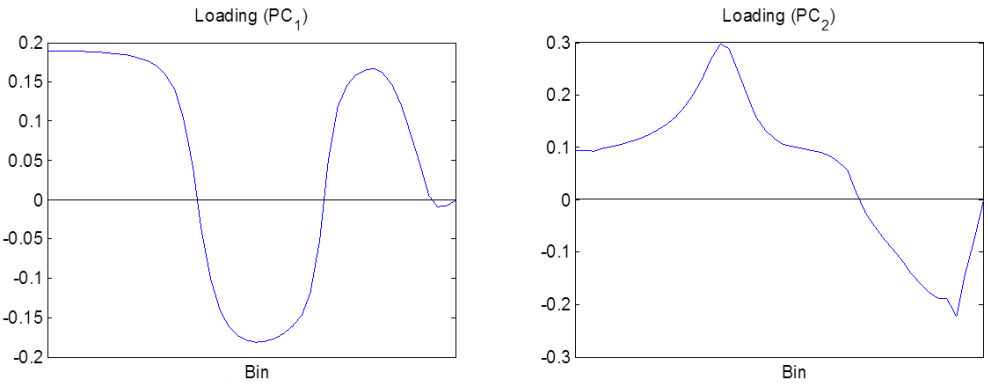

Figure 3.4: Loadings of Principal Components

T^2 and Q-statistics can identify faulty batch as it occurs; however, the Q-statistics indicate many more false alarms compared to the Hotelling's T^2. The Q-statistics show more false alarms because retaining only two principal components only captures about 80% of the variance. If a third principal component is retained, the false alarm rate will decrease; however, both Hotelling's T^2 and Q-statistics are inspected for batch monitoring, and a faulty batch usually can be indicated by both control charts.

3.2.4 Local Continuous Monitoring

Similar to the local batch monitoring, PCA can be applied to the variables in the continuous side of the process in order to reduce the number of variables and monitor unusual events. Key variables in production such as pressure and temperature are monitored. This will be discussed more in detail in the next section, when both the batch and the continuous sides of the

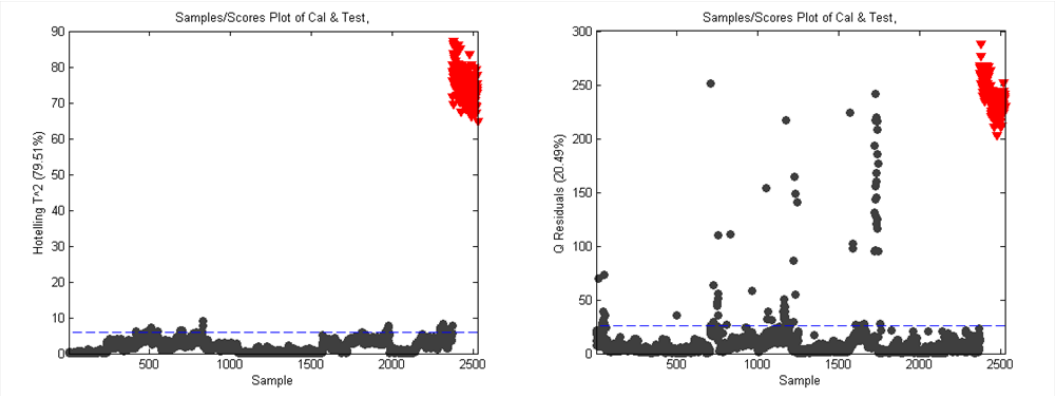

Figure 3.5: Hotelling's T^2 and Q-Statistics

process are modeled together.

3.2.5 Sequential Batch-Continuous Modeling

Modeling for industrial system follows the diagram shown in Figure 3.6. First, the goal and the scope of the model has to be defined. Using process knowledge, the inputs and the outputs of the model have to be determined. This step is critical, as the resulting PLS model should focus on physically meaningful ane measurable inputs and outputs. After that, historical data have to be collected. The inputs need to have proper range of excitations to improve signal-to-noise ratio. As discussed in the previous chapter, the data has to be cleaned and filtered in order to obtain accurate models. In sequential batch-continuous process, the batch data has to be properly aligned as discussed in the previous chapter. After that the data-driven model can be developed and the model validation follows.

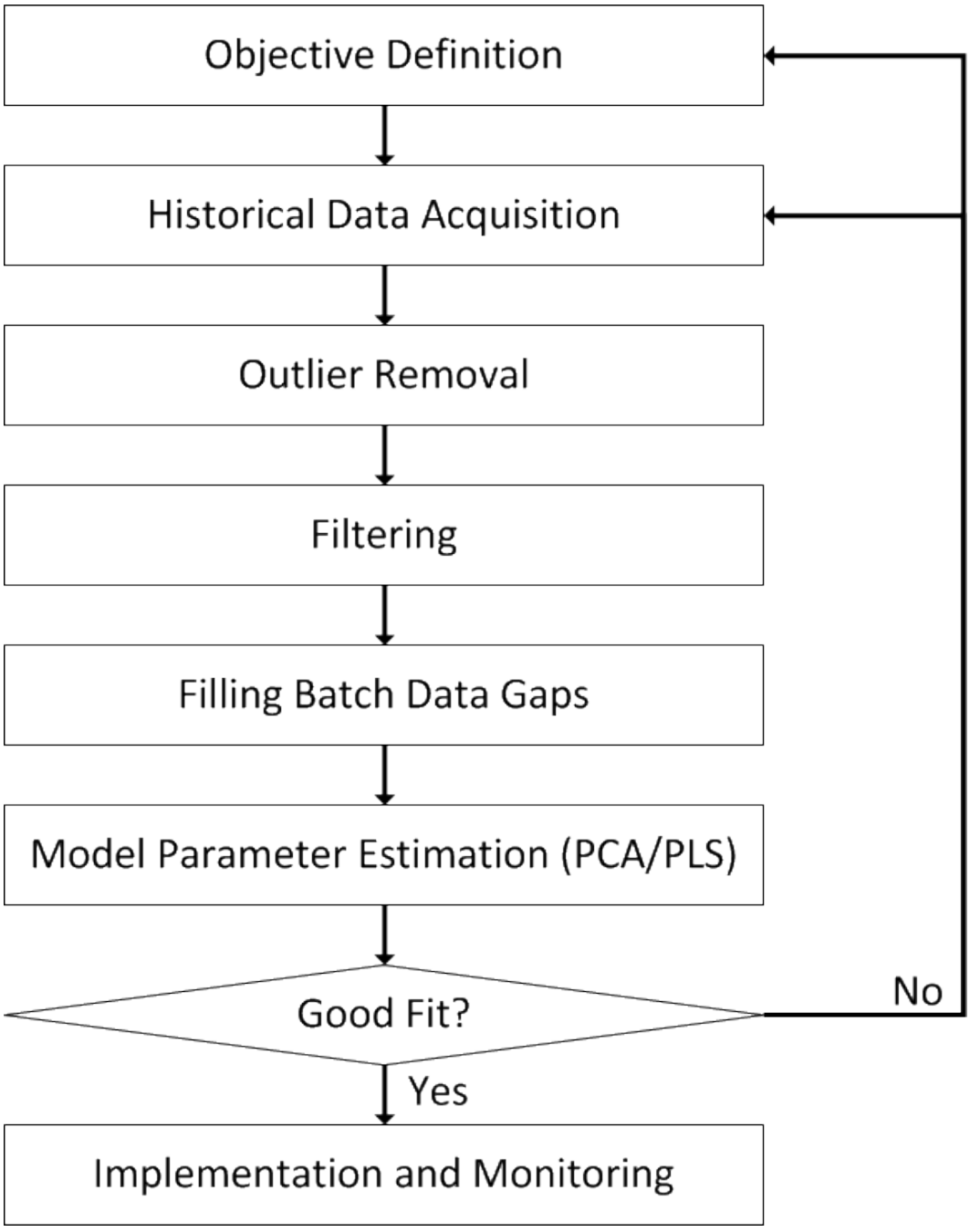

Figure 3.6: Flow Diagram of Industrial Process Modeling

For the industrial system, the goal is to determine the quality of the product that exits the continuous section using data from both the upstream batch and the downstream continuous processes. As shown in Figure 3.7, the upstream batch process has its local monitoring in place that inspects the distribution of the raw materials, followed by the local monitoring in continuous process, which examines the status of the continuous system. The pressure at the exit of the continuous section is chosen to represent the product evolution, while the distribution data from the upstream batch process and the process variables from the downstream continuous process are used as inputs. The data set used for modeling includes more than fifteen different production runs that took place over three years, and therefore accounts for seasonal variations in ambient conditions. More than eighty variables are recorded and the sampling frequency is in the order of seconds. The resulting training data set has more than one million data samples.

3.2.5.1 Dealing with Multiple Operating Modes

In this industrial system that produces different grades of product by varying set-points or inputs, there exist multiple steady-state operation modes [88]. Owing to the inherent nonlinearity of the system, it is unlikely that a single linear model can be used to capture the system behavior in this entire operating space.

In this section, we discuss different approaches for taking multiple operating modes into account. Specifically, multiple linear models are devel-

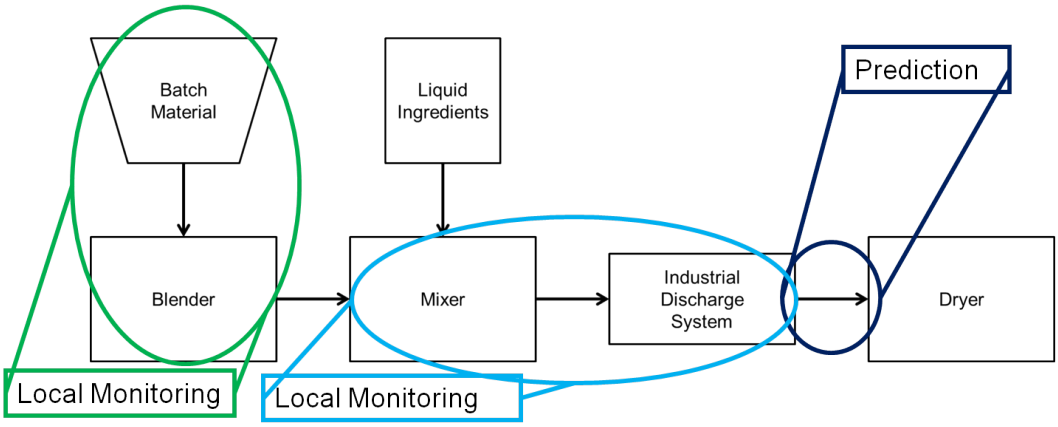

Figure 3.7: Schematic of Monitoring and Modeling for Sequential Batch-Continuous Process

oped, each covering a subset of the operating conditions; the data set is partitioned accordingly using a clustering approach. We compare the resulting multi-model framework with a single model obtained from the overall data set, demonstrating the superior predictive performance of the former. Liu et al. proposed a similar approach by using a detection scheme called growing structure multiple model system (GSMMS) to partition the dataset into different operating regions and developing local linear models to represent the general nonlinear dynamic systems [82].

3.2.5.1.1 One Global Model We begin by developing one "global" model using the entire historical data set as the training data set. The obvious benefit of this approach is that there is only one resulting model. The model maintenance such as parameter update due to process shift or seasonality is

much easier. Also, online implementation becomes easier because there is only one model to choose from. A PLS model was built using the entire data set. As shown in Figure 3.8, the model fit for the training data set is good with R^2 value of 0.947. In this modeling method, the variability from production run to run or from different operating modes are captured and emphasized. However, when the model is tested using a validation data set from a different production run, the fit is poor (Figure 3.9). We believe that the reason for poor fit is the presence of the multiple operating modes. In this global model, the variability that is accentuated is the difference among the multiple modes; however, the variability that occurs within the campaign is not well captured. Variability within a campaign is small compared to the variability from campaign to campaign, in which context it amounts to "noise," which is largely captured by the latent variables that do not make a significant impact on the amount of variance captured and are typically discarded from PCA/PLS models.

3.2.5.1.2 One Model per One Production Run The second method is to build a new model for each campaign, which results in more than fifteen different PLS models. The inputs and the output are the same as in the previous case. Compared to the previous method, in this approach, the resulting models are numerous, which makes it much more difficult to perform model maintenance. Additionally, online implementation is difficult because there are libraries of models to choose from when new data are received. Determining which model to use is difficult and has to be carried out in a trial

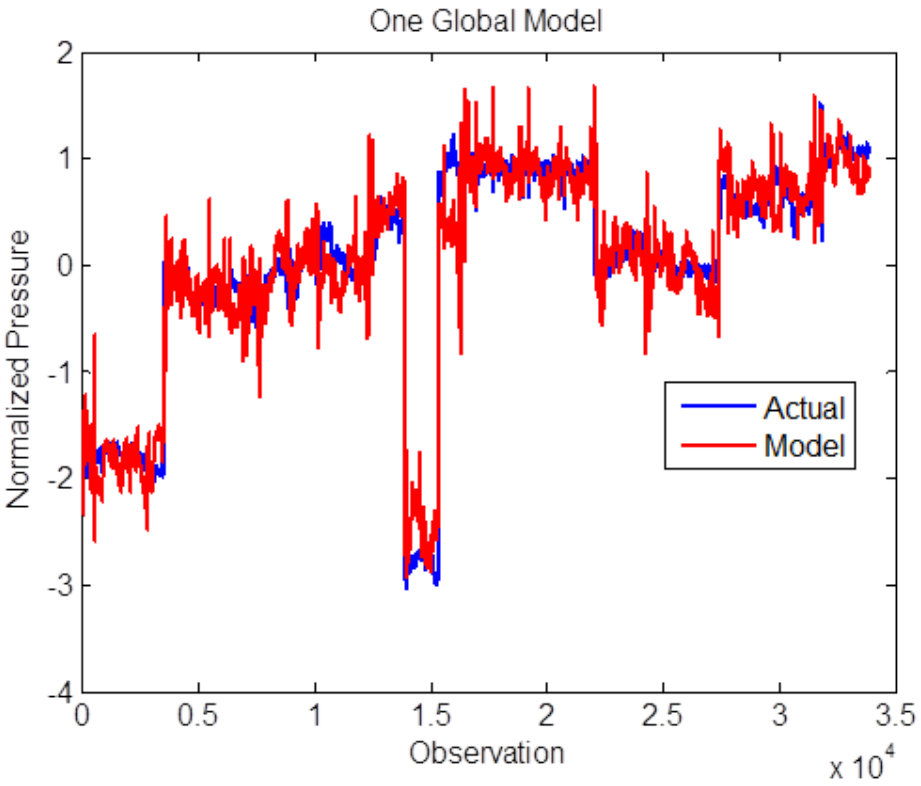

Figure 3.8: PLS Model Fit for One Global Model (Training)

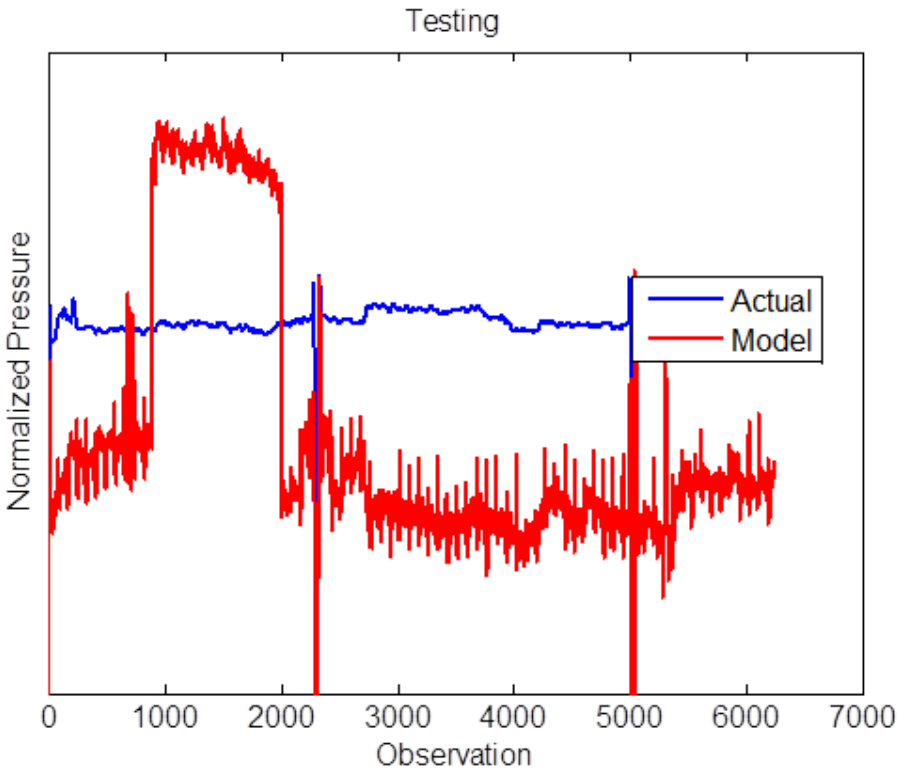

Figure 3.9: PLS Model Fit for One Global Model (Testing)

and error fashion with all existing models. In this approach, the fits for all models for training data sets yielded high R^2 values over 0.9. One model was selected for visualization purposes. As shown in Figure 3.10, the model is able to capture the variability within the model with success and is able to follow the changes of the process that occur during a normal production. However, since the nominal values of the multiple operating modes greatly differ from campaign to campaign, using this current set of models, it is impossible to predict the nominal value for the testing data set (Figure 3.11). Because one production campaign is used to build the model, it is not aware of the differences in the magnitude of output variable from campaign to campaign, thus yielding a poor prediction.

3.2.5.1.3 One Model per One Cluster Finally, we utilize a clustering analysis to divide the global data set into smaller subsets that contain "similar" data. There are many types of clustering methods, which are characterized by how each cluster is defined. For example, k-means clustering, which is used in this application, relies on centroids to determine each cluster. k-means clustering aims to partition the N observations into k sets by minimizing the Euclidean distance between the centroids and the points [87] in the m-dimensional space. The partitioning of the data set is carried out using Equation 3.19.

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{X} \in S_i} \|\mathbf{X} - \mu_i\|^2 \tag{3.19}$$

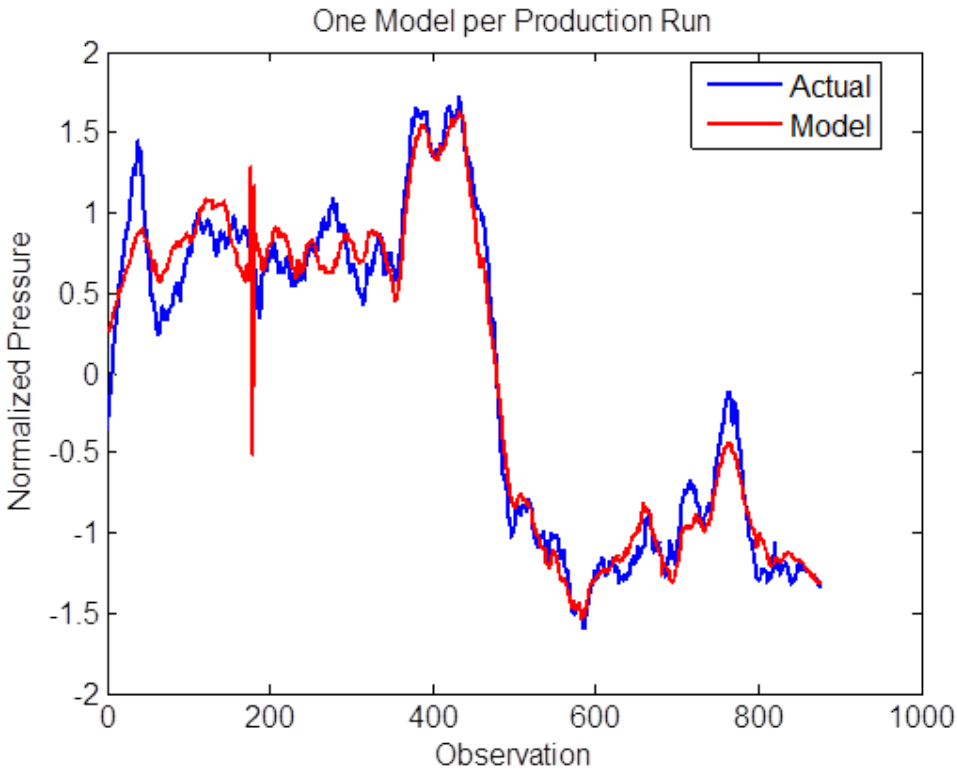

Figure 3.10: PLS Model Fit for One Model per One Production Run (Training)

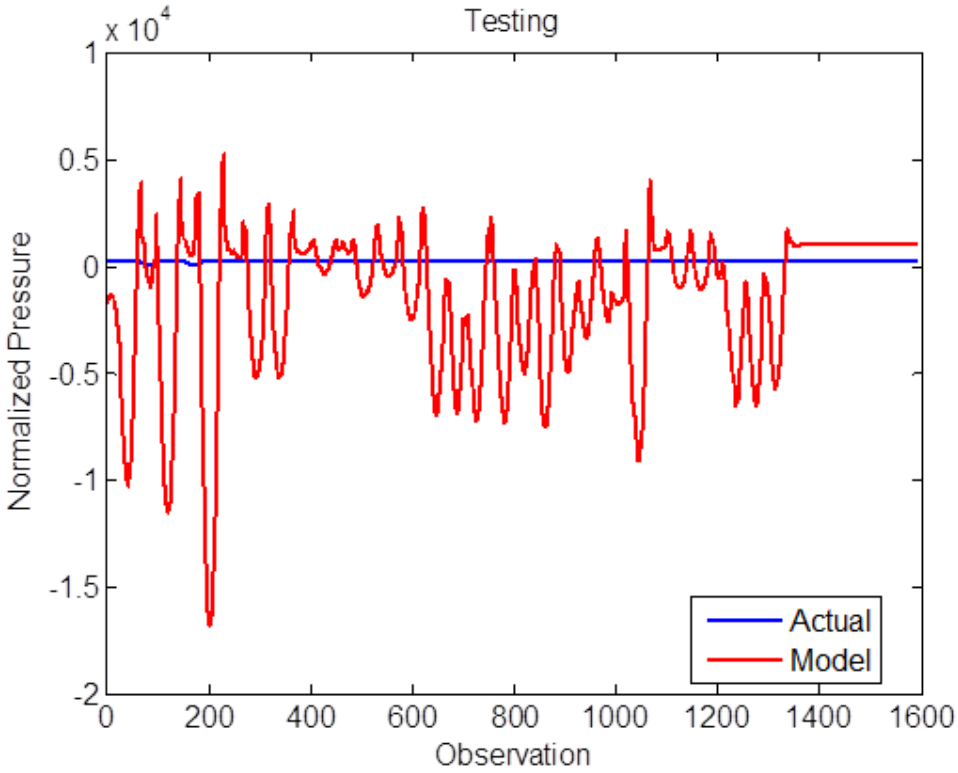

Figure 3.11: PLS Model Fit for One Model per One Production Run (Testing)

$\mathbf{X}$ is a m-dimensional data set with N observations. The k-means clustering algorithm selects random k initial means, μ_i. k clusters are created by associating all the observation with the nearest mean, and the centroid of the cluster becomes the new mean. These steps are repeated iteratively until convergence has been reached. In k-means clustering, the number of clusters, k, needs to be defined a priori. In this analysis, PCA was performed on $\mathbf{X}$ (the process inputs) to reduce the dimension of the data. After PCA application, the PCs were used for k-means clustering. After the clusters were determined, the subsets defined by the cluster analysis were used to build the PLS models. Determining the number of clusters k is a balance between reaping benefits from partitioning versus over-fitting.

First, PCA was performed on the global data set and 10 PCs were retained, capturing 87.5% of the variance. This greatly reduces the dimension of the data set, which relieves the computational difficulty in k-means clustering. The resulting scores were used to find the clusters. In order to determine the number of clusters used, the "elbow" method was used. This method focuses on the sum of all the Euclidean distances from all the data points to their respective clusters. As number of k increases, the distance should decrease; however, at some k, the benefit of adding another cluster becomes marginal. This point is typically determined by visual inspection of a distance vs. number of clusters plot. In our analysis, $k = 8$ was used based on the result in Figure 3.12. Figure 3.13 shows the assignment of the data in the global data set to the eight identified clusters. The figure shows that the all the data

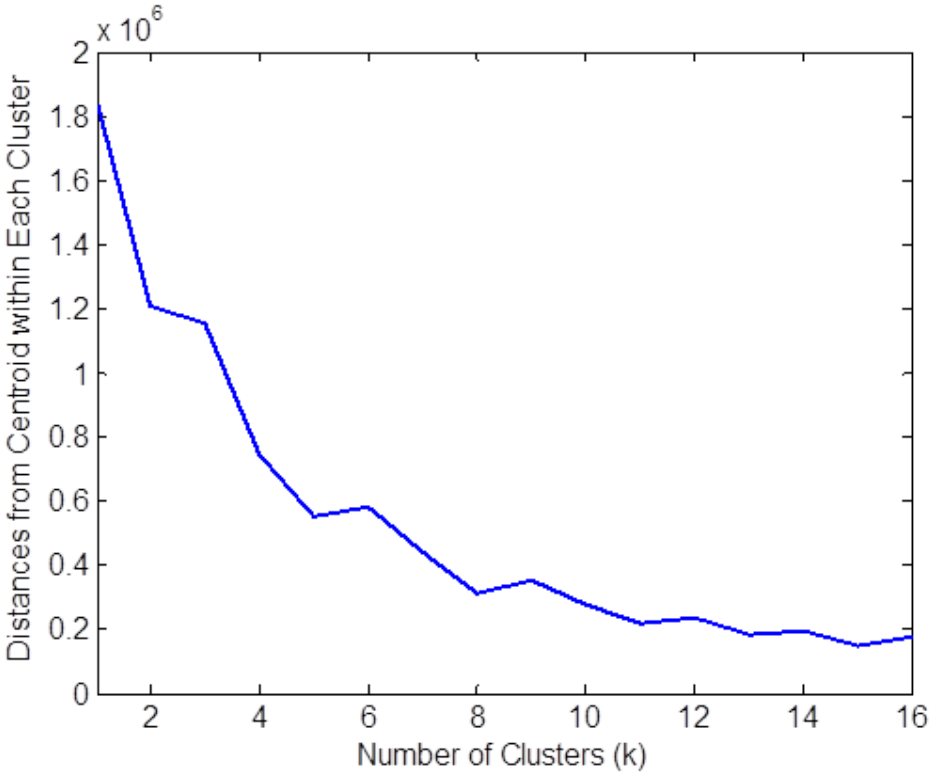

Figure 3.12: Distances from Centroid

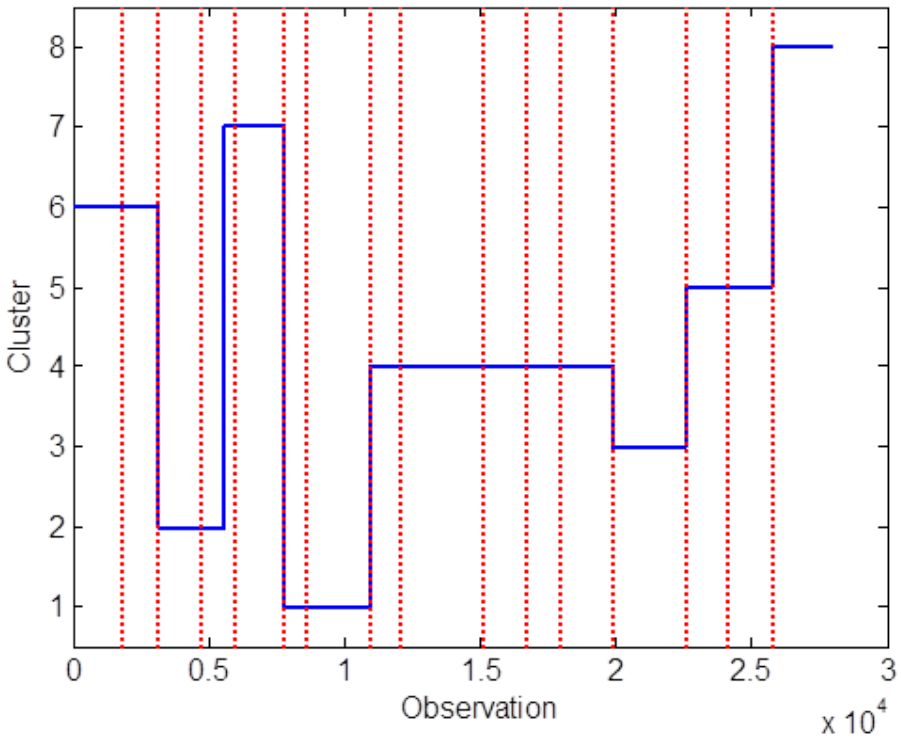

Figure 3.13: Clusters vs. Observations (Red line representing separators of campaigns)

points within each campaign fell into the same cluster (except for one case - see cluster 7), thereby validating our initial assertion that variation within a campaign (and possibly similarities between campaigns) are best captured using multiple separate models. Further note that multiple production campaigns that occurred in similar time frame fell in the same cluster.

Using the results obtained from k-means clustering, eight separate PLS models were created, each having high R^2 values (over 0.9) and fitted output

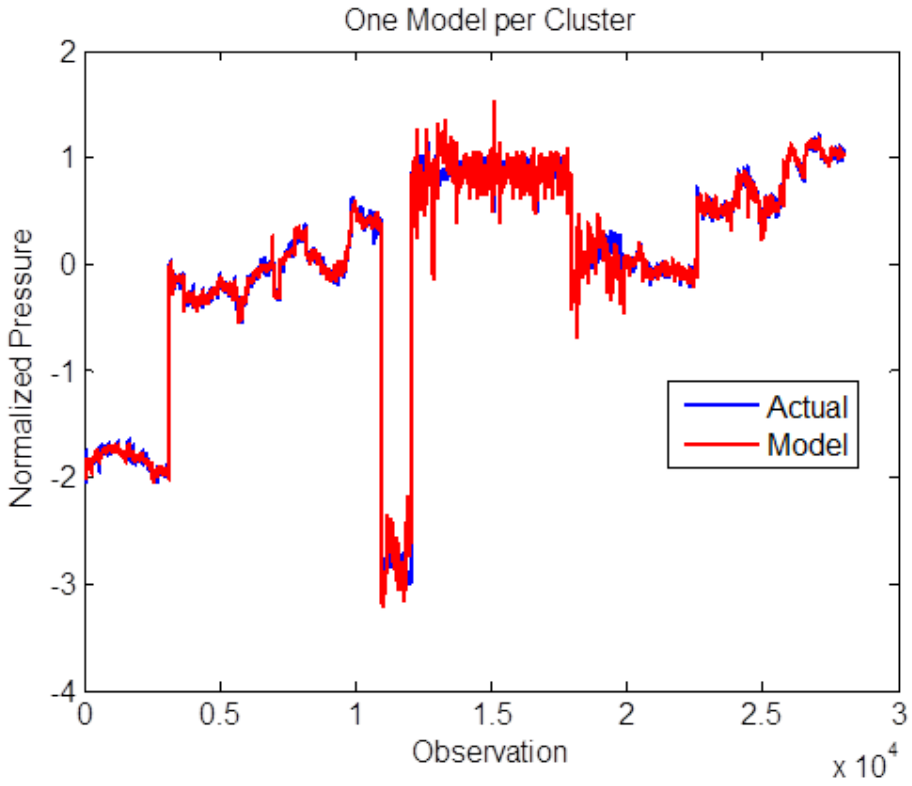

Figure 3.14: PLS Model Fit for One Model per One Cluster (Training)

variable well for training data set (Figure 3.14). The model fit is slightly better than the one global model method as there is no offset in steady-states and the model has less high frequency oscillation. As shown in Figure 3.15, the models can predict the nominal value and variability for the validation data set. Compared to the previous method, determining which cluster and model to use is straightforward because the centroid can be used. When implementing online, the Euclidean distances from all the clusters can be calculated in order to determine which cluster the data falls into.

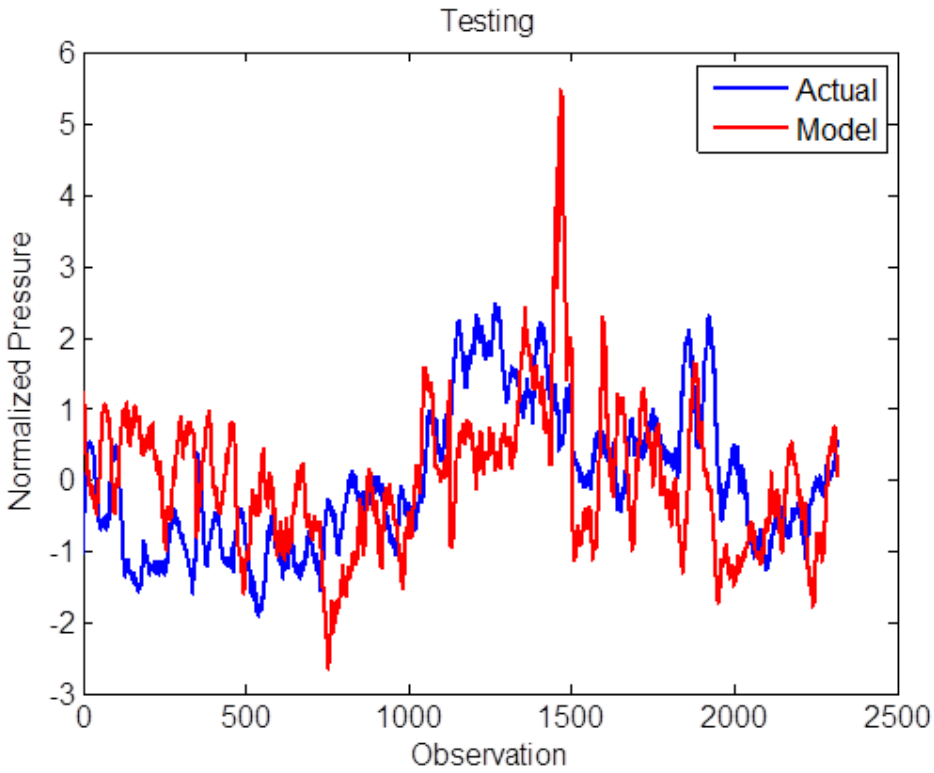

Figure 3.15: PLS Model Fit for One Model per One Cluster (Testing)

3.3 Variable Selection for Modeling

3.3.1 Motivation

The performance of data-driven models derived using PCA/PLS can be improved early on by selecting a subset of *relevant* variables to be used at the model-building stage. The improvement comes from removal of irrelevant, noisy or unreliable variables. Intuitively, this may lead to models with better fit, better predictive ability and more interpretable results [4].

In the chemical industries, several types of variable selection techniques are used to improve the model accuracy and to choose relevant inputs. The general idea is to evaluate a model and the variables using variable importance metrics, rank the variables, and remove variables that are deemed unimportant. Different ranking metrics, such as Variable Importance in Projection (VIP) or normalized beta coefficients, have been proposed. Wold et al. used VIP as the selection metric to determine the subset variables retained [128]. The results of VIP analysis are typically easily understandable and easy to implement. Similar to VIP filtering, normalized beta coefficients from PLS model can be used to determine the variables retained. Since the data are scaled, if the normalized coefficient is close to zero, that variable is not contributing to the PLS model. Fernández et al. utilized backward variable selection (BVS), where they start out with the whole data set, build a PLS model, eliminate the worst variable, and repeat this process until no significant improvement can be obtained. This method can be utilized with any metric (VIP or beta coefficients) [48]. Lu et al. utilized a moving window concept and VIP into

variable selection (MW-VIP), which incorporates the time dependency of the process into selecting relevant variables [85]. Centner et al. developed Uninformative Variable Elimination (UVE) which calculates the cut-off threshold by using the magnitude of noise [20]. Cai et al. built upon this work and developed Monte Carlo UVE (MCUVE) to eliminate some tuning parameters which are difficult to determine for UVE [17]. In this study, we will compare and contrast these different variable selection methods to determine which can be best suited for the chemical industry data, specifically the B2C process under consideration.

3.3.2 Methods

Prior to discussing the details of the methods introduced above, we define the ranking metrics. Beta coefficients can be obtained by fitting a PLS model, which is shown in Equation 3.11. The VIP, which measures the weight of the variable j compared to the rest of the variables can be calculated as follows:

$$VIP_j = \sqrt{K \frac{(\sum_{a=1}^{A} w_a^2 SSY_{comp,a})}{SSY_{cum}}} \tag{3.20}$$

where K is the number of variables, A is the total number of components, w_a is the PLS weight for component a, $SSY_{comp,a}$ is the sum of squares of $\mathbf{Y}$ explained by component a, and SSY_{cum} is the total output variance.

1. **VIP Filtering** VIP filtering introduced by Wold et al. calculates VIP

for each variable and uses this as the decision metric for variable selection [128]. This method utilizes the fact that the average of VIP is 1 because the sum of squares (SS) of all VIP values is equal to the number of variables in $\mathbf{X}$. This means that if all the variables had the same contribution to the PLS model, they will all have VIP values of 1. In order to filter less relevant variables, the VIP values are sorted in descending order and the cut-off value is determined by using process knowledge and inspecting the VIPs of the variables. A widespread choice is to use the "greater than 1" rule, with the underlying statement that variables with VIP values higher than unity are more relevant than variables with sub-unitary VIP values. Similarly, practitioners utilize beta coefficients or a combination of VIP and beta coefficients to eliminate irrelevant variables. If the data set is auto-scaled, the variables with beta coefficients close to 0 have small or little impact to the model. Due to its simplicity and computational efficiency, this method has been the most popular and preferred method in variable selection. However, in real process data, determining the cutoff value is less straightforward. Depending on the input variables, it might not be easy to determine the cut off value for VIP without proper process knowledge. Also, the existence of collinear variables with high correlation lower the VIP value of other variables.

2. **Backward Variable Selection** Backward Variable Selection (BVS) is a method proposed by Fernández et al. [48]. The starting point is the

full data set, for which the model is refitted once the the least important variable is eliminated. This iterative process is repeated until the desired number of variables is reached. The lowest VIP or beta coefficient values are used to determine the least important variable. In this method, the final number of variables is a tuning parameter, which is itself difficult to determine and empirical arguments coupled with process knowledge are typically used. In addition, this method requires iterative evaluation of the ranking metrics, which might result in local optimum. In our comparison, rather than determining the final number of variables, we use "elbow" visualization of R^2 and Q^2 and monitor for significant drop-off in these metrics as a consequence of removing each variable.

3. **Uninformative Variable Elimination** Uninformative Variable Elimination (UVE) developed by Centner et al. uses the reliability of the beta coefficients to determine which variables to retain [20]. The reliability is calculated as follows:

$$c_j = \frac{\beta_j}{s(\beta_j)} \quad j = 1, \ldots, p \tag{3.21}$$

where β_j is the beta coefficient of the jth variable, and $s(\beta_j)$ is the standard deviation of the coefficient. Since the $s(\beta_j)$ cannot be computed directly, Centner et al. propose a "jackknifing" method, in which they acquire the vector of n β_{ij}. They use the average of β_{ij} for β_j and compute the $s(\beta_j)$ as follows:

$$s(\beta_j) = \left(\sum_{i=1}^{n} \frac{(\beta_{ij} - \beta_j)^2}{n-1} \right)^{1/2} \tag{3.22}$$

After computing the reliability, the cutoff rule of $abs(c_j) < abs(max(c_{artiff}))$ is used to determine the uninformative variables. In order to do this, Centner et al. inject an artificial random variable to determine the cut-off value, since the artificial random variable should not be included [20]. This method is simple and does not require any iterations; however, determining the magnitude of the artificial random variable is a tuning parameter with uncertainty, since in real process data the magnitude of the noise for the process variables is different.

4. **Monte Carlo Uninformative Variable Elimination** Cai et al. supplemented existing UVE methods with the use of Monte Carlo sampling method to remove uncertainties in determining the magnitude of artificial random variable [17]. In MCUVE, from the total data set with N observations, a subset with $N_t < N$ observations is used for training the PLS sub-model. This subset is sampled in a Monte Carlo fashion L times. The resulting vector of M β_{ij} can be used to calculate the average, mean(β_j) and the standard deviation, std(β_j). The stability can be calculated in a similar fashion to the reliability, which is given in Equation 3.23.

$$s_j = \text{mean}(\beta_j)/\text{std}(\beta_j) \tag{3.23}$$

After calculating stability, the number of variables retained N_j has to be determined. This is done using an "elbow" analysis based on Root Mean Squared Error of Prediction (RMSEP) using the following equation:

$$\text{RMSEP} = \left[\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2\right]^{1/2} \tag{3.24}$$

where $\hat{y}_i$ and y_i are the predicted and measured value of the ith observation and N is the total number of observations. Choosing N_j entails finding a balance between loss of information and model performance. If the number of retained variables is too small, the robustness and accuracy of the model decrease due to the loss of informative variables. On the other hand, if the number of retained variables is too large, uninformative variables affect and contaminate the model, which results in poor performance.

In our study, RMSEP of the model is calculated for different sets of variables to determine the optimal number of variables similar to the BVS method. There are some tuning parameters that need to be set, such as N_t, L, and N_j. N_t is choosing what percentage of the original data set is used for training set, L is the number of sub-models, and N_j is the number of variables retained. N_t is suggested to be between 40% and 60%, which is setting aside large portion of the training data for validation. With computational power, L can be chosen to be a high value. Large number of MC samples yields more stable mean and

standard deviation values in beta coefficients. Lastly, N_j is determined using the "elbow" analysis.

Table 3.3.2 summarizes the ranking metrics, tuning parameters, advantages, and disadvantages of the variable selection methods discussed in this section.

Method	Ranking Criteria	Tuning Parameter	Advantage	Disadvantage
VIP Filtering by Wold et al. [128]	VIP	Cut-off value for VIP (or "greater than 1" rule)	Simple, interpretable result, widespread	Requires process knowledge, collinear variables bias the ranking
Beta Coefficient Filtering	Beta coefficient	Significance level for beta coefficients	Same as VIP filtering	Same as VIP filtering
Backward Variable Selection (BVS) by Fernández et al. [48]	VIP or beta coefficient	Number of retained variables in the final model	Simple, easily implementable, interpretable result, variable elimination based on each iteration	Tuning parameter greatly affects the final result, iterative evaluation or ranking metrics, could get stuck in local optimum
Uninformative Variable Elimination (UVE) by Centner et al. [20]	Reliability of beta coefficient	Magnitude of artificial random variable	Simple, does not require iteration, provides new ranking metric	Determining tuning parameter is difficult in practice, collinear variables bias the ranking
Monte Carlo UVE (MCUVE) by Cai et al. [17]	Stability of beta coefficient	Percentage of the original data set that is used for training, Number of sub-models, Number of retained variables in the final model	Interpretable results, use of RMSEP can lead to better model fit, collinear variables can be handled due to MC sampling	Many tuning parameters, requires computational power, requires many model evaluations

Table 3.2: Overview of Variable Selection Methods in Section 3.3

3.3.3 Comparison of Variable Selection Methods on Industrial Process

In order to compare these methods for the industrial system, initially over forty different variables were included in the whole data set (note that this data set is slightly different from the one used in section 3.2.5. This is due to the fact that the data used in this section were used to build the soft sensor described in the next chapter). These variables were used to build a PLS model that predicts one product quality variable, which indicates the final product quality. The original data set was partitioned into three different subsets using k-means clustering and three different PLS models were built. The goal is to remove unnecessary variables that might not contribute or even contaminate the model prediction while maintaining or improving model accuracy. When applying the VIP filtering method, the cutoff value of VIP was set as one. When applying the BVS method, VIP was used as the variable ranking metric and the cutoff value was determined using the "elbow" analysis with R^2X and R^2Y values. Similarly when applying the MCUVE method, VIP was used as the variable ranking metric, the cutoff value was determined using the "elbow" analysis with RMSEP values, N_t was set as 40%, and M was set as one thousand.

Prior to discussing the comparison between these selection methods, the regression parameters to evaluate the model need to be discussed. First one is the coefficient of determination, known as, R^2, which can be obtained using Equation 3.25. Residual sum of squares (RSS) is the sum of the squared

difference between the actual observation y_i and the model prediction $\hat{y}_i$. Total sum of squares (TSS) is the total variance that a model can explain which is the sum of the squared difference between the actual observation and the average observation. Another parameter is Q^2 which signifies the predictive ability of the PLS model. This was calculated using Equation 3.26. Predictive error sum of squares (PRESS) is the sum of the squared differences between the actual observation and the response predicted by the regression model. $\hat{y}_{i/i}$ indicates the model estimation when the ith sample was left out from the training data set.

$$\begin{aligned} RSS &= \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \\ TSS &= \sum_{i=1}^{n}(y_i - \bar{y})^2 \\ R^2 &= 1 - \frac{RSS}{TSS} \end{aligned} \tag{3.25}$$

$$\begin{aligned} PRESS &= \sum_{i=1}^{n}(y_i - \hat{y}_{i/i})^2 \\ Q^2 &= 1 - \frac{PRESS}{TSS} \end{aligned} \tag{3.26}$$

Using the VIP filtering method, R^2X, R^2Y, and Q^2 were compared after the variable selection to determine how much of the model has been impacted by the change in variables. The results are summarized in Table 3.3.3. The PLS model of cluster 1 eliminated 23 variables but the model fit

Table 3.3: Model Change after VIP Filtering

Model Fit Metric	Cluster 1	Cluster 2	Cluster 3
R^2X	0.096	0.322	-0.323
R^2Y	-0.046	-0.031	-0.168
Q^2	-0.001	-0.016	-0.091
k	-23	-24	-28

Table 3.4: Model Change after BVS

Model Fit Metric	Cluster 1	Cluster 2	Cluster 3
R^2X	0.082	0.294	0.077
R^2Y	-0.013	-0.006	-0.008
Q^2	0.016	0.025	0.017
k	-17	-22	-12

does not alter too much (R^2Y only decreases by about 5%). A similar result is obtained for the PLS model of cluster 2, but it appears that in the case of cluster 3, variable elimination may have been too aggressive since model fit is degraded.

The results obtained using the BVS are summarized in Table 3.3.3. This method, as mentioned above, requires iteration, which means that in order to eliminate k variables, $k + 1$ number of different PLS models had to be fitted. The "elbow" as shown in Figure 3.16 is used to determine the final number of variable retained. As discussed earlier, as the number of variables removed increases, the model accuracy and performance decrease due to loss of information.

Lastly, using MCUVE requires the most computational power among all these methods, as it requires randomly creating numerous subsets of the

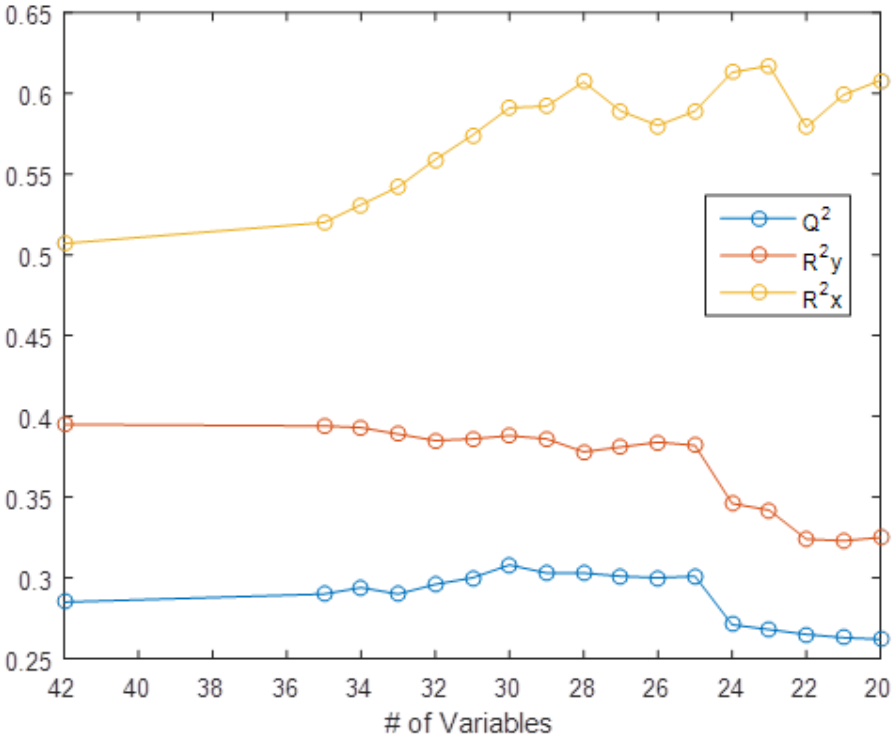

Figure 3.16: Model Fit Metrics vs. Number of Variables Eliminated (BVS)

Table 3.5: Model Change after MCUVE

Model Fit Metric	Cluster 1	Cluster 2	Cluster 3
R^2Y	0.016	0.010	0.002
k	-24	-22	-17

whole data set and building many PLS models to calculate the stability of each variable. Compared to VIP filtering and BVS, the results after MCUVE for cluster 1 and 2 are fairly similar. The number of variables eliminated is similar and the changes in R^2 might be negligible. However, the variable selection method for cluster 3 using VIP filtering was too aggressive and using BVS might have been too lenient. As shown in Table 3.3.3, using MCUVE, the PLS model does not lose information and model accuracy yet we can remove 17 variables. In order to determine the number of variables eliminated, MCUVE utilizes RMSEP to determine the optimal number of variable retained. As shown in Figure 3.17, as number of variables eliminated increases, the RMSEP decreases as the uninformative variables that might contaminate the model with noise are removed. At a certain point the RMSEP starts increasing - this means that variables with information required to accurately model for $\mathbf{Y}$ are being eliminated.

Figure 3.18 shows the variables were eliminated or retained from the forty two initial variables. A red box means that the variable was removed and a blue box means that the variable was retained. Some of the variables such as variables 27, 28, 32 and 34 were consistently eliminated while some variables such as variables 13, 24, and 26 were retained in all clusters by all three

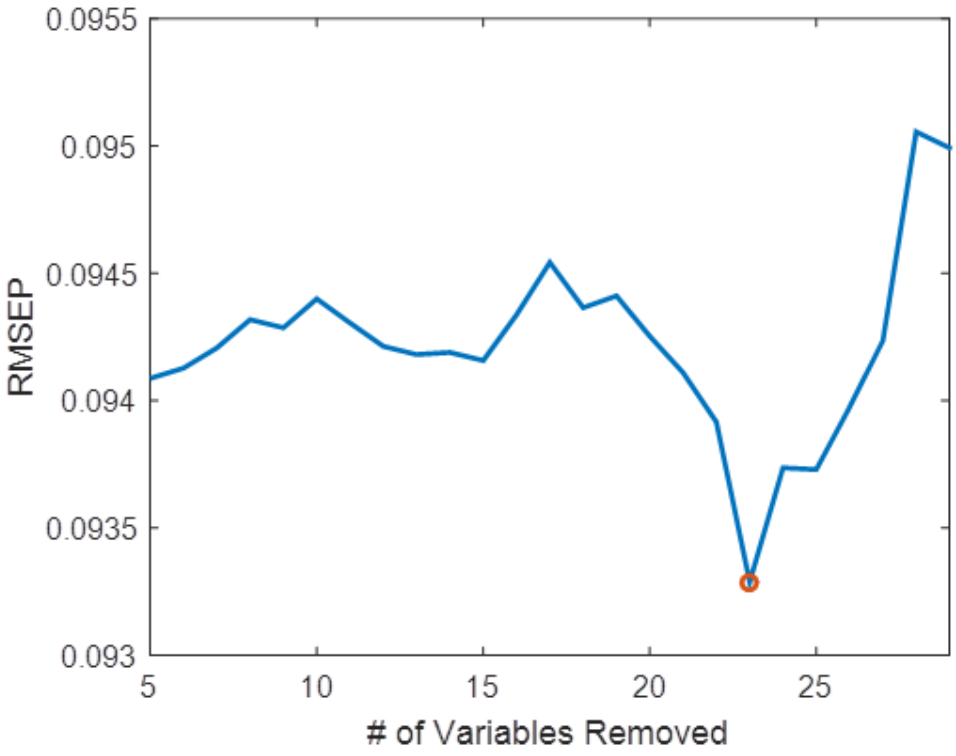

Figure 3.17: RMSEP vs. Number of Variables Eliminated (MCUVE)

methods. As can be seen across the figure, different variable selection methods retain and eliminate variables differently. Moreover, different clusters have different set of variables retained which imply that these multiple production modes are impacted differently.

3.4 Summary

In this chapter, data-driven modeling techniques, which include PCA, PLS are discussed. The multiple operating modes present in complex sequential batch-continuous process were modeled by partitioning the data set into different clusters. Various variable selection methods for PLS are compared.

In order to model and monitor the status of the batch process and continuous process, data-driven modeling techniques such as PCA and PLS are used to capture the variance within the data set while reducing the dimensionality of the data, which facilitates the monitoring. PCA is applied to the distribution data of the batch process and using two principal components, the physical change in the distribution was represented. Faulty batches were successfully identified using the Hotelling's T^2 and Q-statistics. In order to handle multiple operating modes of the industrial B2C system, three different modeling techniques are introduced. Using one global model, the PLS model fit for the training data set yields a good fit, but for the testing data set, the model prediction is poor due to different nominal values associated with multiple operating conditions. By contrast, applying clustering analysis and using ne model per one cluster yields very good results, and online implementation

	VIP Filtering			BVS			MCUVE		
	Cluster 1	Cluster 2	Cluster 3	Cluster 1	Cluster 2	Cluster 3	Cluster 1	Cluster 2	Cluster 3
Variable 1	0	0	0	1	1	1	0	0	0
Variable 2	0	0	0	0	0	1	0	0	1
Variable 3	1	0	0	1	0	0	0	1	1
Variable 4	1	0	0	1	0	0	1	1	1
Variable 5	1	0	0	1	0	0	1	0	0
Variable 6	0	0	1	0	0	1	0	0	0
Variable 7	0	0	1	0	0	1	0	0	0
Variable 8	0	1	0	0	1	1	1	1	1
Variable 9	0	0	1	1	0	1	0	0	0
Variable 10	1	0	0	1	1	1	1	0	1
Variable 11	0	1	0	1	1	1	1	1	0
Variable 12	1	1	0	1	0	0	0	0	0
Variable 13	1	1	1	1	1	1	1	1	1
Variable 14	1	1	0	1	1	1	0	1	1
Variable 15	1	0	1	1	0	1	1	0	1
Variable 16	0	0	1	0	0	1	0	1	0
Variable 17	0	0	0	1	0	0	0	0	0
Variable 18	0	0	1	1	0	1	0	0	1
Variable 19	1	0	0	1	0	1	0	1	1
Variable 20	0	1	1	0	1	1	1	1	0
Variable 21	1	1	0	1	1	1	1	1	0
Variable 22	1	0	1	1	0	1	0	1	1
Variable 23	0	0	0	0	0	1	0	0	1
Variable 24	1	1	1	1	1	1	1	1	1
Variable 25	1	0	1	1	0	1	0	0	1
Variable 26	1	1	1	1	1	1	1	1	1
Variable 27	0	0	0	0	0	0	0	0	0
Variable 28	0	0	0	0	0	0	0	0	0
Variable 29	1	1	0	1	1	1	1	1	1
Variable 30	1	1	0	1	1	0	0	1	1
Variable 31	0	1	0	1	1	1	1	0	1
Variable 32	0	0	0	0	0	0	0	0	0
Variable 33	0	1	0	0	1	1	1	1	1
Variable 34	0	0	0	0	0	0	0	0	0
Variable 35	0	1	0	0	1	1	1	0	1
Variable 36	0	1	0	0	1	1	1	0	1
Variable 37	1	1	0	1	1	1	0	0	0
Variable 38	1	1	1	1	1	1	0	0	1
Variable 39	0	1	1	0	1	1	0	1	1
Variable 40	0	0	0	0	1	1	1	1	0
Variable 41	1	0	0	1	0	0	0	1	1
Variable 42	0	0	0	0	0	0	1	1	1

Figure 3.18: Variables Eliminated Using Different Variable Selection Methods

is simplified by using the centroids of the clusters to select the operating region of the process and the corresponding model.

In addition to the modeling methods, different variable selection methods were compared to eliminate irrelevant variables that might contaminate the PLS model with noise. All the variable selection techniques use some ranking metrics such as the magnitude of the beta coefficients of VIP. Various methods have different tuning parameters, which may be difficult to select without process expertise. VIP filtering is the most prevalent method with simple and interpretable result using VIP. VIP filtering, however, could be too aggressive or lenient depending on the cutoff value. Usually, "greater than 1" rule is used, which is a good starting point, but might not be sufficient for real industrial data. BVS is easy to implement and yields similar result to VIP filtering, but could get be affected by local optima. MCUVE improves UVE by using Monte Carlo sampling to remove uncertainties in setting some tuning parameters which might be difficult in real process data. The "elbow" empirical evaluation is used to determine the number of variables retained which balances between the model accuracy and the loss of information. For the industrial system, MCUVE results in the largest number of variables eliminated while maintaining model accuracy and performance. Reducing the number of variables makes the model more robust, as measurements could be jeopardized by malfunctioning sensor or contaminated measurement.

Chapter 4

Real-Time Optimization of Sequential Batch-Continuous Process

4.1 Introduction

In this chapter, we focus on developing a soft sensor model for a quality variable that is expensive to measure and improving the process operation using RTO. In the first section, a simple method to correlate the product quality variable with low measurement frequency to continuously measured process variables. Because each measurement is costly (in the sense that a sample taken from the production line is analyzed by a technician and is destroyed in the process), it is often impractical to measure the final quality variable continuously or even frequently. We discuss a way to down-sample the process variables that are measured at higher frequency and develop a soft sensor model which can be used to estimate the ware quality. We use the term "soft sensor" to denote a model-based computation that is used to infer the values of the quality variable at time instants when measurements are not available; the reader is referred to, e.g., [105, 42, 84, 63] for more information concerning soft sensing in the process industries.

In the second section, the soft sensor is used to identify the optimal pro-

cess operating conditions in a real-time optimization (RTO) calculation [114]. The goal is to maintain the product quality within the acceptable boundary while process changes occur.

4.2 Selection of Product Quality Variable

4.2.1 Motivation

In a complex chemical process with many operational steps, it becomes quite difficult and expensive to measure the product quality at intermediate steps. As an example, the cost of a composition analyzer is an order of magnitude (or more) higher than the cost of a temperature or pressure probe. Consequently, product quality measurements are often collected only at the outlet of the process and reflect the final product quality. Further still, and depending on the nature of the process, measuring the final product quality directly is itself challenging (both technically and economically), involving offline "lab" measurements and analyses performed operators or specialized technicians. Incorporating this type of variables into modeling becomes particularly difficult due to inherently low sampling rate and to time delays associated with the intervention and work of operators.

Low sampling rates (typically of the order of hours) make it impossible to correctly reconstruct the process dynamics in soft sensor formulation, and confine the operation of the soft sensor to predicting steady-state values of the quality variable. An associated challenge is the selection of appropriate data for calibrating the sensor against process measurements. Specifically, as

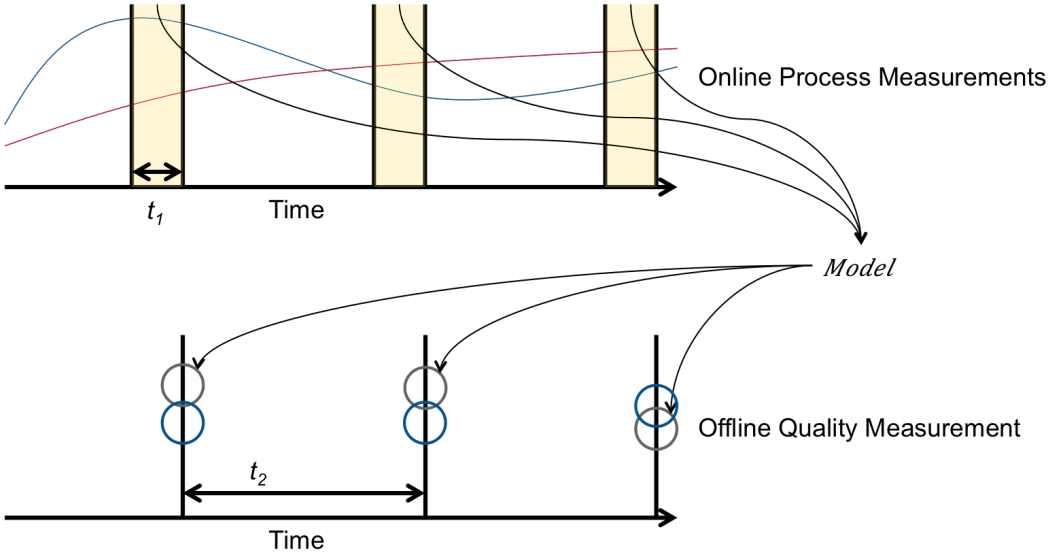

Figure 4.1: Schematic Representation of Soft Sensor

shown in Figure 4.1, one must choose the time interval prior to the quality variable measurement, from which data are used as the *sensor input*. This window must be, i) sufficiently long to capture the operation of the process at steady-state, yet, ii) short enough to ensure that the quality variable reading is correctly associated with the recent evolution of the process rather than reflect long-term trends.

4.2.2 Methods

As shown in Figure 4.1, the online process variables are measured continuously as indicated by blue and red lines while the offline quality variables, shown as blue circles, are measured offline intermittently (sampling frequency of $1/t_2$). t_1 is used to determine the current state of the process. Because

the real process data contains high measurement noise, it is difficult to obtain clean a variable trajectory even after data preprocessing such as outlier removal and filtering, thus it is suggested to use the average or median value in window t_1 rather than using a single point. t_1 acts as a tuning parameter as large window size (high t_1 value) would cover process change while small window size might still be influenced heavily by process noise. In case of developing a steady-state model such as a PCA or PLS model, the model fit can be improved by removing observations in process transients or in dynamics. The standard deviation of the process variables in window t_1 may be used as a metric to determine whether the process is in a transient state or at steady-state. As shown in Figure 4.2, when at steady-state, the standard deviation value should only capture the measurement noise; on the other hand, when in transition, the standard deviation value is affected by dynamics of the process. Choosing a proper value for t_1 becomes crucial and requires some process knowledge in order to differentiate process dynamics from measurement noise. In order to develop a steady-state model, the observations in transition and dynamics need to be discarded as they may contaminate the model.

In the industrial system, the operators collect samples from the line and measure the quality properties of the samples. Measurement errors, measurement bias due to shift changes, destructive loss of product for testing are a few disadvantages of the current approach. In order to model the system output, we follow the concepts outlined above, by down-sampling the variables for which continuous/frequent measurements are available to a sampling

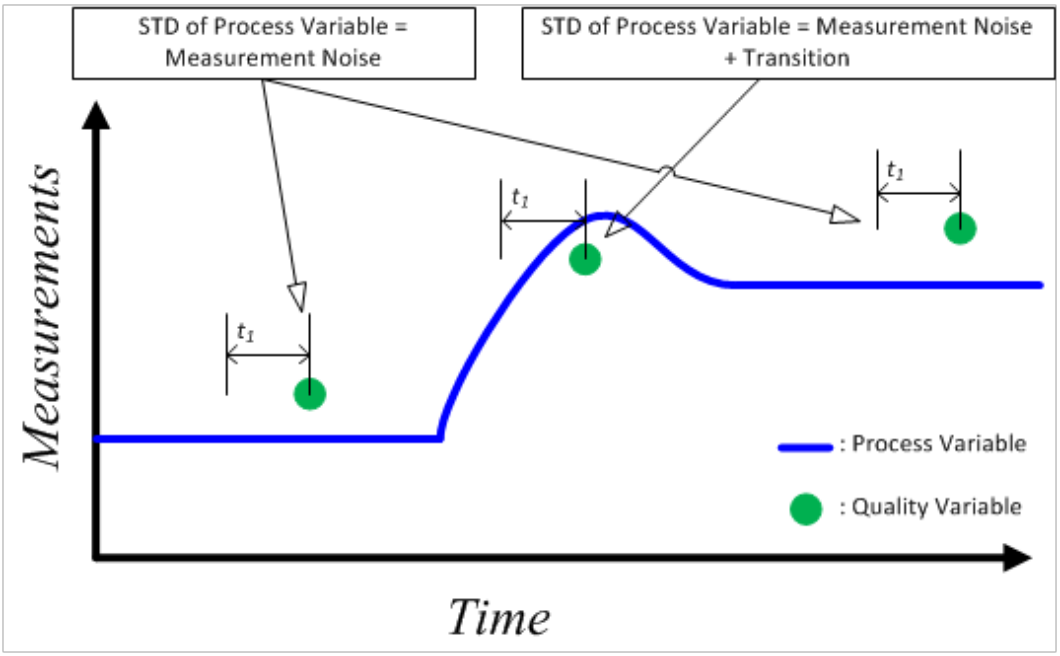

Figure 4.2: Schematic Representation of Soft Sensor

frequency on par to that at which measurements of the quality variable are collected, i.e., once every few hours.

In order to develop a soft sensor for the industrial process, a linear PLS model was developed using data which span over twenty separate production runs. Within the data set, twenty eight different inputs were used as inputs to the model to fit the output quality variable. Prior to fitting a model, t_1 was determined to be sixty. Figure 4.3 shows the actual measurement taken offline by the operators and the PLS model estimation. Note that the number of observations is low compared to that of Figure 3.14. The general trend and different steady-state nominal values exhibit good fit, but the actual measurement contains high frequency noise. This could be due to lack of information from inputs and higher noise level of process measurements compared to that of offline quality measurement. In the following section, we utilize the soft sensor developed for RTO calculation in order to find the optimal sequence of manipulated input to operate the process more efficiently.

4.3 Real Time Optimization (RTO)

4.3.1 Problem Formulation

The goal of the industrial system is to maintain the final product quality within specifications in the face of upstream disturbances. The final product qualities such as shape, hardness, stiffness are affected by many different factors such as raw materials distribution, amount of liquid agent added to the batches, pressure, temperature, and torque in the system. The system

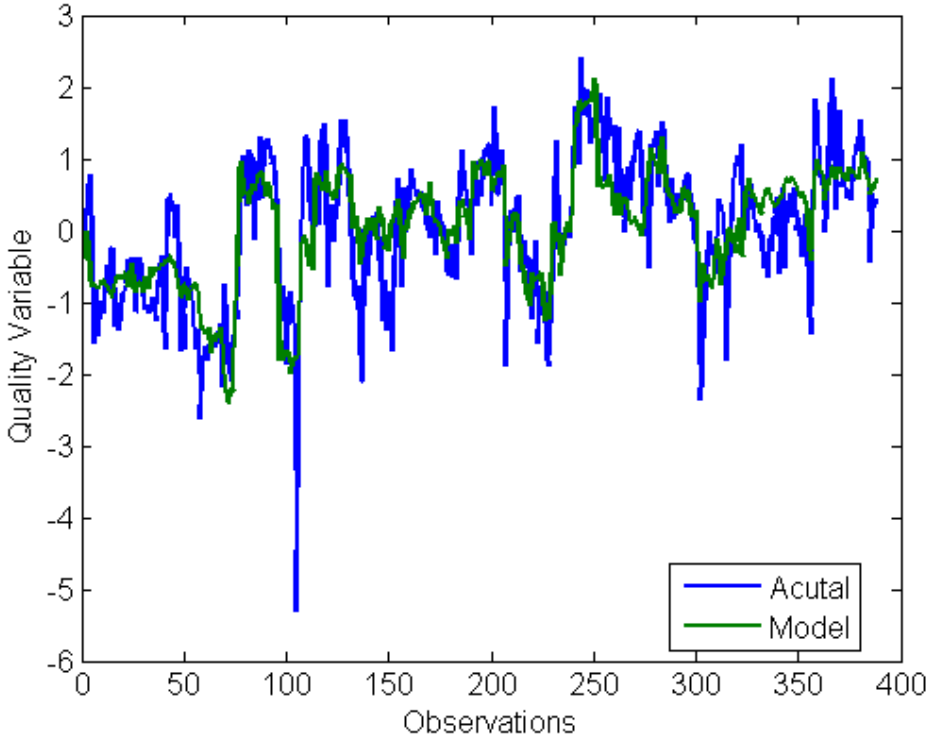

Figure 4.3: Schematic Representation of Soft Sensor

is currently operated under manual control, whereby operators take quality measurements and determine empirically the appropriate amount of liquid that needs to be added to the process.

The availability of the steady-state PLS model described above opens the possibility of performing this calculation in an automated fashion and in closed-loop. This entails carrying out two steps: first, the quality variable is estimated using immediate past historic measurements, then, ii) the RTO calculation is performed to determine the flow rate of liquid to be added to the process for the period leading to the next estimation and RTO calculation.

This approach amounts to a control vector parameterization, whereby the manipulated variable (liquid carrier flow rate) is approximated via a piecewise constant function; equivalently, the liquid carrier flow rate remains constant between two RTO calculations, with the time interval between these calculations being a tuning parameter.

The objective of the RTO calculation is to find optimal amount of liquid added to the process that maintains the quality of the final product within specification in the face of disturbances and process changes. It is important to note that the steady-state PLS models developed (Figure 4.3) are subject to model uncertainties, which need to be accounted for in the following optimization. The problem 4.1 represents the optimization formulation that drives the process to the setpoint.

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & (y_{SP} - y)^2 \\ \text{subject to} \quad & y = \mathbf{X}\beta' + u\beta'_u \\ & lb \leq u \leq ub \end{aligned} \tag{4.1}$$

y is the product quality estimation from the model, y_{SP} is the predetermined setpoint for the product, β and β_u are the PLS model coefficients, $\mathbf{X}$ is the PLS model input, u is the manipulated variable (amount of liquid agent added to the system), and lastly, lb and ub are the lower and upper bounds for the manipulated input, respectively. It is a quadratic program (QP) that calculates the optimal amount of liquid added to the process while closely following the predetermined setpoint. Alternatively, formulation 4.2 can be used for less aggressive approach. Formulation 4.1 might set aggressive movements for u due to the objective function penalizing any moment that y does not match y_{SP}. On the other hand, formulation 4.2 has the in-spec parameter $p_{threshold}$ which acts as a buffer. The magnitude of $p_{threshold}$ is a tuning parameter that can be utilized for tighter operation versus relaxed burden on the manipulated input. In order to choose a proper value for $p_{threshold}$, some process knowledge on the bounds for the acceptable products is required.

$$
\begin{aligned}
\underset{u,p}{\text{minimize}} \quad & p \\
\text{subject to} \quad & y = \mathbf{X}\beta' + u\beta'_u \\
& lb \leq u \leq ub \\
& y_{SP} - y \leq p \\
& y - y_{SP} \leq p \\
& y \geq 0 \\
& p \geq p_{threshold}
\end{aligned}
\tag{4.2}
$$

Unlike (4.1), problem 4.2 is a linear program (LP), which can be solved very efficiently. This problem formulation aims to ensure that the product quality remains within a range $p_{threshold}$ of its desired setpoint/value, with last constraint allowing for deviations from this threshold if it cannot be achieved.

4.3.2 Results

The overall goal of modeling and optimizing the industrial system is to maintain the product quality within specifications in the presence of disturbances, consisting either of changes in feedstock quality or changes in product type. The information from the actual production data with the product quality measurement data are used to emulate the operation of the industrial system, and the operator heuristics are used to determine the setpoint and the in-spec buffer parameter $p_{threshold}$. Here, y_{SP} is 0 (note that we use deviation variables to camouflage the physical values of the variables), and $p_{threshold}$ is chosen to be 0.1. Lower and upper bounds for the manipulated inputs were

chosen to be the minimum and maximum values from the production data. Industrial data were used to emulate the inputs $\mathbf{X}$.

First, we present the results of using the RTO formulation 4.1. Figure 4.4 (top) compares the actual measurement from production (red) to the simulated results obtained from implementing the RTO (blue). Figure 4.4 (bottom) indicates that implementing the RTO strategy maintains the process output relatively close to the setpoint except from observations 1 to 50, where the system cannot be driven to the setpoint because the manipulated input is hitting the lower bound. The manipulated input does not make any drastic movements but does hit the upper bound around observation 250.

We also present the result from the RTO formulation 4.2. This formulation utilizes the buffer variable to avoid large changes that might occur in the manipulated input which might wear out or damage actuators. Figure 4.5 presents the simulation results. Compared to the previous case (using (4.1)), the quality variable fluctuates within the in-spec band of $y_{SP}, \pm p_{threshold}$. The changes in manipulated input are far less pronounced, with the upper bound on the flow rate never becoming active. Qualitatively, these results reproduce more closely the data secured from the plant, and suggest that operator preferences and decisions tend to be conservative and relatively accepting of slight deviations from quality prescriptions. We conjecture, however, that the closed-loop implementation of an automatic control/optimization system based on (4.1), rather than the "milder" formulation (4.2), will lead to improved economic benefits due to more consistent product quality, and we are

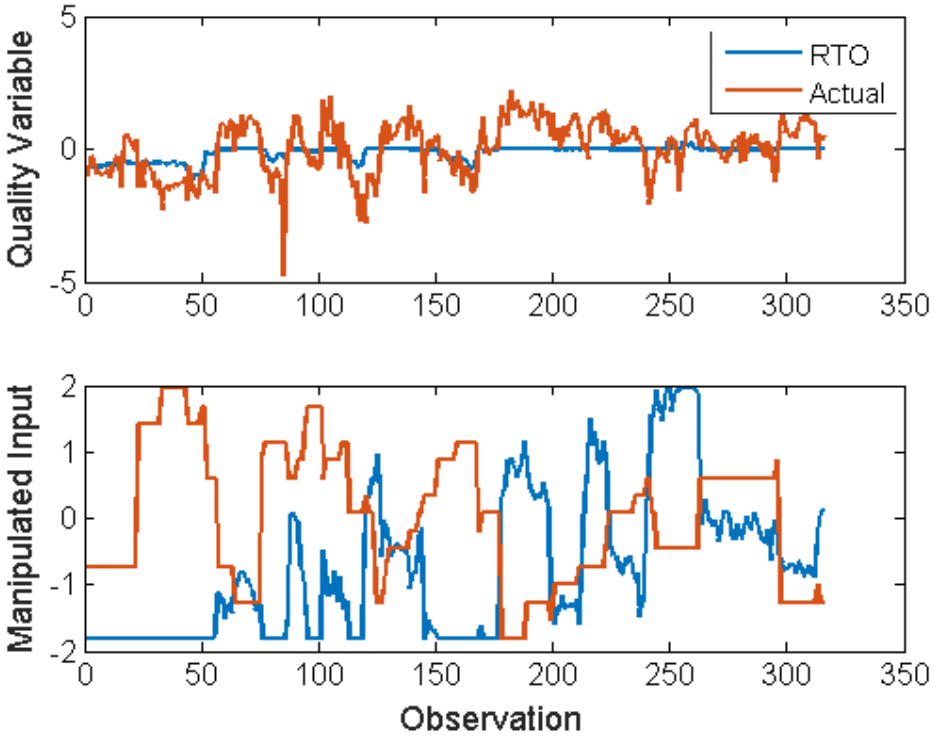

Figure 4.4: RTO Result from Problem Formulation 4.1. The actual measurement from production are shown in red, and the real-time optimization calculations are shown in blue.

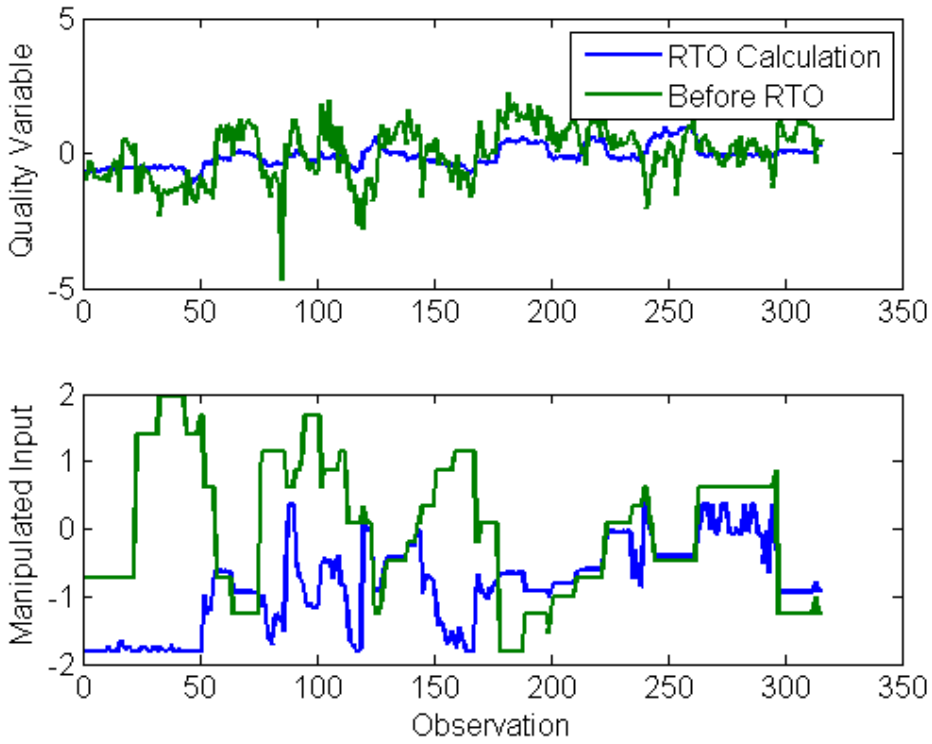

Figure 4.5: RTO Result from Problem Formulation 4.2. The actual measurement from production are shown in green, and the real-time optimization calculations are shown in blue.

hopeful that this conjecture can be validated in real-life plant operations in the near future.

The product specification variable p is shown in Figure 4.6. The RTO result only violates the given specification a few times around observation 1, 50, 110 and 150, compared to constant violation of product specification before the RTO calculation was carried out. Also, the instances that RTO produces out-of spec products can be alleviated by changing the lower bound of the

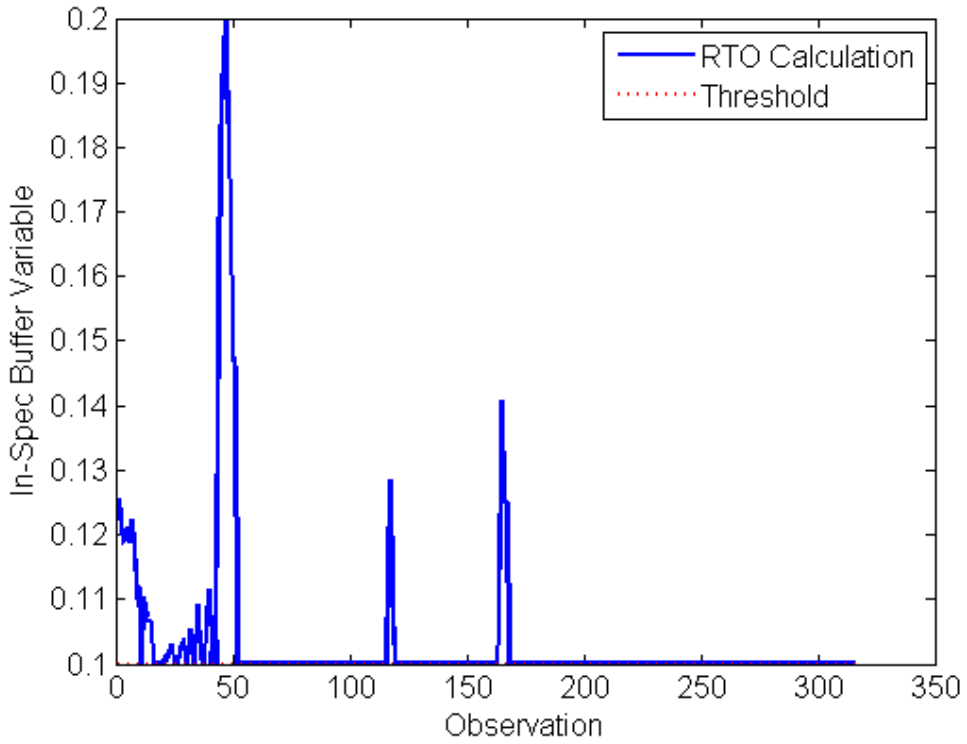

Figure 4.6: Product Specification from Problem Formulation 4.2

manipulated input.

4.4 Summary

In this chapter, the development of a soft sensor for a product quality variable which is determined via infrequent lab sampling is discussed. The sensor is presented in the form of a steady-state PLS model, correlating the product quality variable to a set of variables that are measured online and in real time. The soft sensor is used to develop a real-time optimization formu-

lation, aimed at maintaining product quality within specification by altering a manipulated input of the process within given bounds.

The chapter presents two main contributions: a down-sampling technique for synchronizing the data collected in real time with the lab-sampled measurements, and a "buffered" formulation of the real time optimization problem, which is expressed in terms of an easy to solve linear program.

Validation on data collected from an industrial system proves that these concepts can be beneficial for industrial operations, and are encouraging for pursuing practical implementation.

Chapter 5

Integration of Scheduling and Control

5.1 Introduction

Scheduling and control are two essential functions in the decision-making of the chemical supply chain, dealing with the common goal of maximizing profit from operations by setting production targets based on demand and ensuring that the targets are met in the presence of process disturbances and operational uncertainty.

Over the past decades, the importance of production scheduling and capacity planning has been emphasized as a necessity towards maximizing economic benefits at the enterprise level [52]. Significant advances have been made in process control, particularly as far as advanced, supervisory control techniques are concerned [8]. It is anticipated that further economic benefits can be derived from a tighter coordination of all levels of decision-making in a chemical enterprise and, in particular, from a closer integration of production scheduling and process control [11].

At the fundamental level, production scheduling and supervisory process control utilize the same framework: solving an optimization problem constrained, amongst other, by the equations of a process model; intuitively, an

integrated formulation of these two activities should be quite natural.

Nevertheless, the integration of scheduling and control faces several challenges, both human [116] and technical. Elaborating on the latter, we note that scheduling and control systems make and implement their decisions in different time scales [55]. Scheduling horizons range from several days to weeks, while control systems compute their decisions with minute frequencies and considering time horizons that are typically a few hours long (Figure 5.1).

Moreover, scheduling calculations are carried out under the (often implicit) assumption that the process operates (mostly) at steady state, and that the transitions between products and/or operating states i) are short, compared to the periods of steady-state operation, and, ii) can be characterized in terms of the tabulated transition times, typically a set of time-invariant parameters. Under these circumstances, scheduling problems are formulated as mixed-integer linear programs (MILPs) that are agnostic to process dynamics beyond capturing the tabulated transition times.

On the other hand, control calculations must account for the process dynamics, and, fundamentally consist of solving a dynamic optimization problem (which is typically formulated in a discretized form to facilitate numerical solutions).

The integration of scheduling and control must merge the aforementioned process representations; the resulting (integrated) problem is thus most likely a mixed-integer dynamic optimization (MIDO) that becomes extremely

difficult to solve in a practical amount of time. Several factors contribute to this: first, detailed dynamic process models are often nonlinear, high dimensional and stiff. Second, the general problem formulation is infinite-dimensional and must be discretized in time. In turn, the discretization (typically, a MINLP) must consider a long time horizon (to account for scheduling needs) with a small time step (that takes into consideration the dynamic evolution of the process). Third, the problem is mixed-integer in nature, with integer variables used to reflect scheduling decisions (e.g., production sequence, product assignment to slots) and continuous variables corresponding, e.g., to the states of the dynamic process model.

The need to address these challenges has prompted developments in two main directions. On the one hand, a top-down approach advocates incorporating control considerations into a scheduling framework, while bottom-up approaches aim at extending formulation of supervisory control to account for scheduling considerations [11].

1. **"Top-Down" Approach** These studies have focused on improving the link between control and scheduling by relating the transition times used in (otherwise static) scheduling calculations to the process dynamics in order to optimize control performance [89, 32]. Some researchers formulated the scheduling and control problem as a large mixed-integer dynamic optimization (MIDO) over the entire production cycle [2, 21, 50, 51, 92, 104].

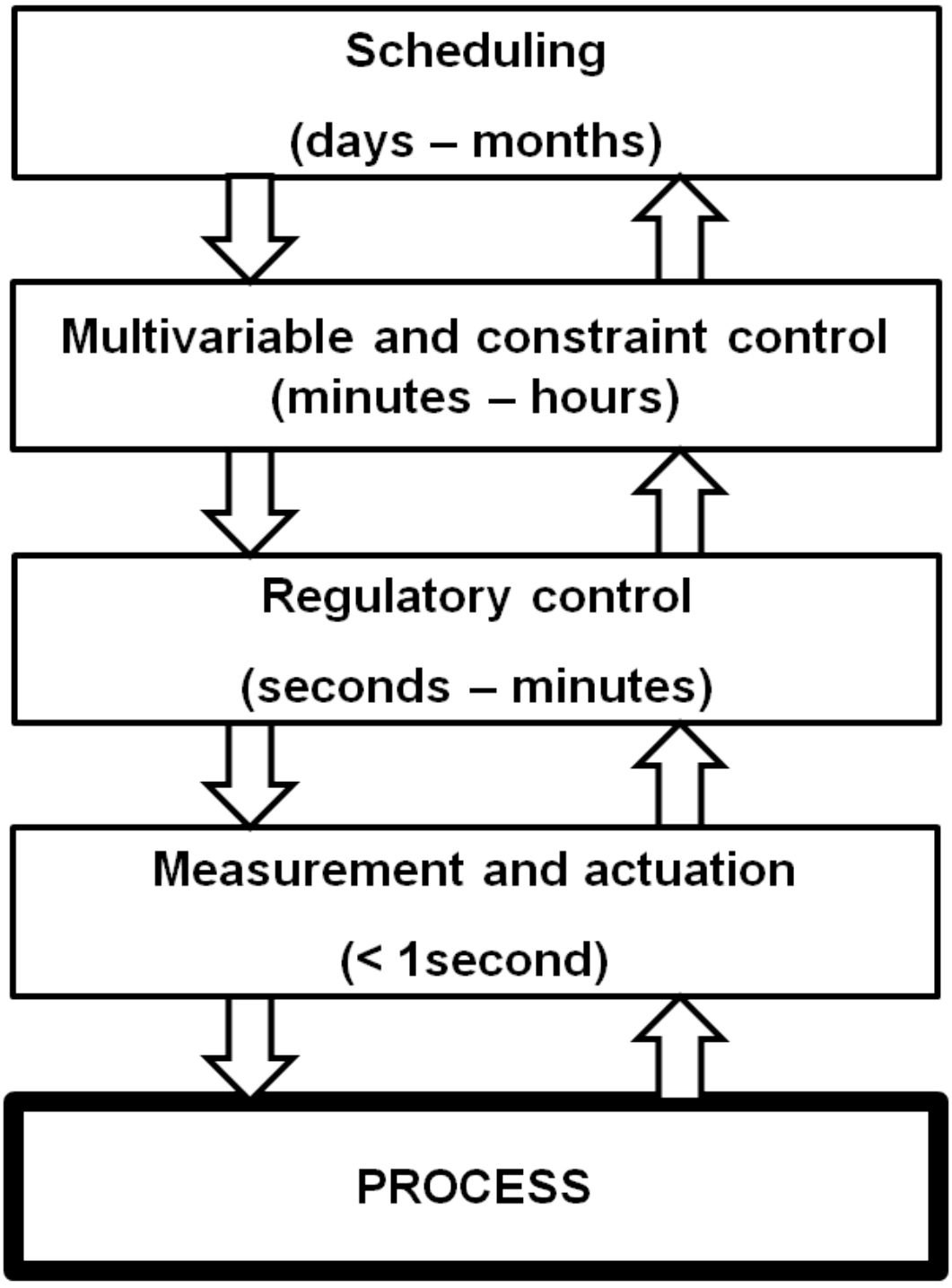

Figure 5.1: Hierarchy of Decision Making in the Chemical Supply Chain [114]

2. **"Bottom-Up" Approach**

   In this case, the scheduling objective and constraints are considered in the level of supervisory controller, which leads to the extension of economic model predictive control (EMPC) [46, 47, 56, 62, 3].

It is also worth mentioning the important strides made in numerical methods aimed at solving this class of problems [124, 123, 15].

In this chapter, we introduce a new approach of the top-down category; namely, we investigate the use of a time scale bridging model (SBM) for integrating short-term scheduling and supervisory control. The SBM is a low-dimensional dynamic model that captures the closed-loop input-output behavior of the plant [41, 99, 13], that is, the dynamics of the process and its controller. More specifically, the SBM provides an explicit description of the transient evolution of scheduling-relevant process outputs (e.g., product quality, production rate) as a function of changes in the corresponding set-points/targets, as computed by the scheduling layer.

The SBM is then embedded in the scheduling calculation, thereby providing information on the closed-loop dynamics of the process *without* the need to consider the entire process model.

Our approach is grounded in past research concerning the dynamics of process systems [11, 8, 7, 121, 122, 33, 61, 10, 12, 9], where it was shown that the plant-wide, scheduling relevant dynamics of the aforementioned type

evolve over a longer time horizon that typically far exceeds the response time of individual unit operations in a process flowsheet.

5.2 Problem Definition

We concentrate on a continuous process capable of producing multiple products from the same raw materials by altering processing conditions. The production schedule is assumed to be cyclical, and that any product can be produced only once in a production cycle. Once one cycle is ended, the identical cycle follows. Lastly, it is assumed that the product price, inventory cost, and the demand for each product is deterministic and known a priori.

The process dynamics can be represented as follows:

$$\begin{aligned} 0 &= f(\dot{x}, x, z, u) \\ 0 &= g(x, z, u) \end{aligned} \tag{5.1}$$

where x represent state variables, u represent input variables, and z represent algebraic variable, making this set of differential algebraic equations (DAE). Each product is defined in terms of a state x of the system, and all the products are produced at different and unique steady-states. We also assume that any steady state can be reached from any other steady state, i.e., that there are no "banned transitions" in the operation of the process.

5.2.1 Scheduling

From a scheduling perspective,the objective is to identify the optimal production schedule (in terms of the order in which products are produced, and the amount of time expended on producing each product) that meets the product demands while reducing the inventory cost. The problem is set in a continuous time framework, where the production makespan T_C is divided into a number of "slots" of duration T_j. The decision variables include the production sequence (i.e., the allocation of products to each production slot), the length of these slots, and the makespan. The objective function incorporates profit and inventory cost, and is written as follows:

$$\max_{T_C,T_j,b_i} \left(J = \sum_{i=1}^{N_P} C_{P,i} P_i - \sum_{i=1}^{N_P} C_{I,i} W_i \right) \tag{5.2}$$

under the following constraints:

- Timing Constraints

$$\begin{aligned} T_C =& \sum_{j=1}^{N_S} T_j \\ T_{S,LB} \leq& T_j \leq T_{S,UB} \\ T_{C,LB} \leq& T_C \leq T_{C,UB} \end{aligned} \tag{5.3}$$

The first constraint simply defines the production makespan, which is the lengths of all the time slots. Index j represents the time slots and N_S is the total number of time slots. The next two constraints set the

lower and upper bounds for the production length and the makespan, respectively.

- Sequence Constraints

$$\begin{aligned} &\sum_{i=1}^{N_P} b_i = 1 \qquad b_i \in [0,1] \\ &N_P = N_S \end{aligned} \tag{5.4}$$

The first constraint states that only one product can be manufactured within one time slot. Index i denotes the product number and N_P represents the total number of products. The number of slots N_S is set to be equal to N_P, which ensures that each product is manufactured only once during the cycle. Finally, the binary variable, b_i, denotes that the product assignment is a discrete decision.

- Demand Satisfaction

$$\begin{aligned} P_i &= \int_0^{T_C} F dt \qquad \text{if } x = x_i \\ W_i &= \int_0^{T_C} P_i dt \\ D_{LB,i} &\leq P_i \leq D_{UB,i} \end{aligned} \tag{5.5}$$

P_i is the amount of on-spec product i manufactured, and W_i is the amount of on-spec product i stored by the end of the production cycle. The product is considered on-spec, if the concentration of the product (or the state of the process) is within the specified condition of the product (x_i). The amount of product manufactured has to be within demand

lower and upper bounds, $D_{LB,i}$ and, respectively, $D_{UB,i}$, which are assumed to be known a priori.

5.2.2 Control

From a control perspective, the goal is to maintain the process at the desired steady-states, and to transition between these states quickly and safely. In the process, the manipulated inputs u are utilized to change the states x, which correspond to products P_i [1]. In order to shift between the products quickly, the system has to be under tight control, which reduces the total makespan.

We define the (state feedback) control law as:

$$u = K(x, x_{SP}, \theta) \tag{5.6}$$

where x_{SP} represent the setpoints (corresponding to different products) and θ are a set of controller tuning parameters.

The process dynamic model (5.1) together with the control law (5.6) can be used to represent the closed-loop dynamics of the process. Note that choosing this representation of the dynamics to embed in the scheduling calculation provides no dimensionality reduction benefit. However, the control law can be chosen such that it provides a well-defined closed-loop behavior, that can be explicitly characterized using a low-order dynamic model. One of the

[1] For simplicity, we rely on the assumption that full state information is available for the process under consideration; a discussion in terms of process outputs was provided in [41]

controller design approaches that satisfy this need is input-output linearization (see, e.g., the works of Kravaris and Kantor [69, 70] for a comprehensive review). Under certain conditions on the structure and dynamics of the system (which typically involve it being "square" - the work by Kolavennu et al. [65] and having stable zero dynamics), input-output linearizing controllers imposed a linear closed-loop input-output behavior of the form:

$$\sum_{j=0}^{r} \beta_j \frac{d^j y}{dt^j} = y_{SP} \tag{5.7}$$

where y is a system output (which in our case is chosen to be one of the states x), and r is the relative order of the system (roughly speaking, r represents the number of times the input is integrated before it affects the desired output).

Equation 5.7 thus constitutes precisely the scale-bridging model that we are after: it provides information regarding the dynamic evolution of y (x in our case) as a function of the forcing function y_{SP}, the setpoint, is low(er)-dimensional than the original model (note that r is at most equal to the dimension of the state space of the system) and, more importantly, linear; these features make this class of models highly appealing for formulating and solving integrated scheduling and control problems.

5.2.3 Integrating Scheduling and Control

Traditionally, scheduling calculations provide the targets for the supervisory control system. The scheduling calculation is carried out separately by

utilizing the predetermined information about transitions between the product.

Here, three solution methods for the scheduling and control problems are contrasted: first, a *static* one, which is aligned with the sequential paradigm above (i.e., it is assumed that transition times are known a priori). Second, we consider the *full dynamic* problem that incorporates the entire process model as an additional set of constraints in the scheduling formulation. Third, we validate the approach proposed above, whereby the full dynamic model is replaced by the scale-bridging model.

5.2.3.1 Static Scheduling

Static scheduling can be simply represented as shown in Equation 5.8. Constraints 5.5 are redefined, in the sense that the transition times between the steady-states are tabulated and assumed to be time-invariant. In this case, the scheduling problem becomes a mixed integer program (MIP). This approach does not truly represent the integration of scheduling and control.

$$\begin{aligned} \max_{T_C, T_j, b_i} \quad & J_{Scheduling} \\ \text{subject to} \quad & \text{Scheduling Constraints} \\ & \text{Predefined Timing Constraints} \\ & y \in D_y \end{aligned} \tag{5.8}$$

5.2.3.2 Scheduling using the Full Dynamic Model of the Process

In this solution approach, the full-order process model is incorporated into the scheduling problem as an additional set of constraints, thus solving the scheduling and control problem simultaneously, as shown in Equation 5.9. As opposed to static scheduling, this approach does not approximate the transition time, rather it is determined by the process model.

In this case, it is required to include the inputs u as decision variables in the optimization problem, and the solution will thus also provide the optimal (open-loop) trajectory of the process inputs, in addition to the optimal makespan T_C, the optimal production time of each slot T_j, and the optimal sequence of products x. This scheduling problem is a mixed integer dynamic optimization (MIDO), because of all the process model, which is significantly more difficult to solve compared to the static MIP approach.

$$\begin{aligned} \max_{T_C, T_j, b_i} \quad & J_{Scheduling} \\ \text{subject to} \quad & \text{Scheduling Constraints} \\ & \text{Dynamic Process Model} \\ & x, y, u \in D_{x,y,u} \end{aligned} \tag{5.9}$$

We note here that this approach presents a major disadvantage from a control perspective, that is, the process input sequence is calculated at the start of the production period for the entire makespan. While this sequence is (locally) optimal in the ideal case (i.e., the model is perfect and there are no operational disturbances), it is unlikely to stay so in practical cases whereby

plant-model mismatch and outside disturbances are inevitable. Furthermore, this can have deleterious effects in the case where the process itself may become open-loop unstable.

5.2.3.3 Scheduling with Scale-Bridging Models

Here, instead of using the (highly nonlinear, high-dimensional, stiff) full-order process model as a constraint, a the SBM (e.g., Equation (5.7)) is used to provide a low order representation of the scheduling-relevant closed-loop input-output behavior of the process:

$$\begin{aligned} \max_{T_C, T_j, b_i} \quad & J_{Scheduling} \\ \text{subject to} \quad & \text{Scheduling Constraints} \\ & \text{Scale-Bridging Model} \\ & y \in D_y \end{aligned} \tag{5.10}$$

This model will be referred to as time scale bridging model (SBM). We define the model as the explicit function relating the supervisory controller setpoint to the measured process outputs that are of interest to scheduling of the form shown in Equation 5.7.

Time scale bridging approach therefore proceeds similar to full dynamic scheduling, with the exception that the full process model is replaced with the time scale bridging model, and the decision variables that are related to process manipulated variables are replaced by the setpoints of the supervisory controller. The solution thus consists of the optimal setpoint sequence that

imposes, via the supervisory controller,

$$y_{SP}(t) = \sum_{i=1}^{N_P} y_i^{SS} b_i(t) \tag{5.11}$$

where $y_{SP}(t)$ represents the setpoint to be tracked by the process output y over the makespan. y_i^{SS} is the desired product operating condition of product i. The optimal production sequence depends on the binary decision variable, $b_i(t)$, which indicates the choice of products i over the makespan.

The use of SBM significantly reduces the numerical complexity of the scheduling calculation, as highly nonlinear process model is replaced with a linear (system of) ordinary differential equation(s). The resulting MIDO is thus likely less demanding form a computational point of view than the problem (5.9).

5.3 Numerical Solution Approach

Two different approaches can be taken in order to solve the integrated scheduling and control problem. These methods include the simultaneous approach and the sequential approach. The simultaneous approach requires discretization of the state and the control profile in time, which in essence reformulates MIDO into a large scale mixed integer nonlinear program (MINLP). This approach fully discretizes the differential algebraic equations (DAEs) and does not rely on any DAE solvers. Instead, they are handled with the NLP solver. This, however, means that it requires an efficient efficient NLP

solver that can handle a large system and also requires first and second order derivatives in order to carry out optimization. Flores-Tlacuahuac et al. introduced the simultaneous approach to solving a cyclic scheduling problem for a multi-product CSTR [50, 51]. Terrazas-Moreno et al. used the simultaneous approach to find optimal sequence of a polymerization reactor [119]. These aforementioned approaches calculate the entire schedule and control action off-line and implement the calculated setpoint, and thus are disadvantageous in disturbance rejection [92].

In the sequential approach, three different components (the (MI)NLP solver, the DAE solver, and sensitivity calculations) are required. In each iteration of the optimization, decision variables are defined by the (MI)NLP solver. These decision variables are used to solve the DAEs over the specified time horizon, and then sensitivities are integrated together with the DAE system to obtain the gradients of the objective function and the constraints with respect to the decision variables, which are then used by the (MI)NLP solver in the subsequent iteration in order to update the (MI)NLP solver. The sequential approach requires less function and gradient evaluations compared to the simultaneous approach; however, the performance and computational efficiency of the sequential approach depend on the DAE solver and the sensitivity calculations. Allgor et al. proposed a generalized decomposition method for MIDO problems [2]. Chatzidoukas et al. decomposed a MIDO into a master problem and a dynamic optimization dual problem to obtain the optimal production schedule and the optimal grade transition profile between steady

states [21]. Prata et al. combined disjunctions and logical constraints with differential algebraic model for a continuous polymerization process, and solve a MIDO to generate the optimal production policy [104].

We demonstrate through two case studies - 1. a multi-product continuously stirred tank reactor (CSTR) and 2. a process network consisting of a multi-product CSTR with external heat exchanger, which makes extensive use of energy recovery in order to decrease utility costs. These process models are stiff and highly nonlinear especially the second case study.

5.4 Case Studies

5.4.1 Multi-product CSTR

We consider a non-isothermal multi-product CSTR with four products manufactured at different operating conditions. The developments presented in this subsection follow very closely the results reported in the paper [99]. Our goal is to maximize overall production profit while meeting the manufacturing demand for each product. The scheduling problem thus consists of determining the total production time, the optimal production sequence and the processing times for each product. We solve this problem following the three approaches described above, i.e., static scheduling, full dynamic scheduling, and dynamic scheduling using an internal coupling model. The full dynamic scheduling problem for this system has been formulated and solved by Flores-Tlacuahuac et al. and we follow closely their developments in that direction, as well as using the same model parameters as in their paper [50]. The static

scheduling problem is solved assuming a constant transition time $\tau = 10hr$ for all transitions.

Let us now focus on the development of the SBM for this process. The process model is given in Equation 5.12 and 5.13, where y_1 is the dimensionless concentration, and y_2 stands for the dimensionless temperature.

$$\frac{dy_1}{dt} = \frac{1 - y_1}{\tau} - k_{10}e^{-N/y_2}y_1 \tag{5.12}$$

$$\frac{dy_2}{dt} = \frac{y_f - y_2}{\tau} + k_{10}e^{-N/y_2}y_1 - \alpha u(y_2 - y_c) \tag{5.13}$$

In Equation 5.12, τ represents reactor residence time, k_{10} is the pre-exponential factor, and N is the activation energy. In Equation 5.13, y_f denotes the dimensionless feed temperature, y_c is the dimensionless coolant temperature, α is the dimensionless heat transfer area. The coolant flow rate u is the manipulated variable available for changing the output of interest, which in this case is the composition y_1. The relative order of this system is $r = 2$. We design a nonlinear input-output linearizing controller with integral action to impose a critically damped second-order input-output behavior:

$$\tau_{CM}^2 \frac{dy_1}{dt} + 2\tau_{CM}\frac{dy_1}{dt} + y_1 = y_1^{SP} \tag{5.14}$$

with $\tau_{CM} = 2hr$ (note that this value was chosen so that the step response of the closed-loop system reaches steady state in about $10hr$, which is comparable to the transition time used for static scheduling. The SBM-based scheduling is

Table 5.1: Optimal Solutions to Three Different Scheduling Formulation Problems

Case	Profit	Production Sequence	Total Production Time
SBM	36.559	$3 \rightarrow 1 \rightarrow 2 \rightarrow 4$	$119.749hr$
Static	36.615	$2 \rightarrow 1 \rightarrow 3 \rightarrow 4$	$119.585hr$
Full	35.455	$4 \rightarrow 2 \rightarrow 1 \rightarrow 3$	$122.785hr$

Table 5.2: Optimal Solution for SBM-based Scheduling

k	1	2	3	4
τ_k	10.06	10	10	10.104
i	1	2	3	4
W_i	35.100	21.525	20.330	27.526
$tp_{i,k}$	$(i = 3, k = 1)$	$(i = 1, k = 2)$	$(i = 2, k = 3)$	$(i = 4, k = 4)$
	18.35	16.064	15.171	30

thus formulated using the above equation as a constraint, aiming to determine the optimal setpoint profile.

5.4.1.1 Optimal Solutions to Scheduling Problem Formulations

The static, full dynamic and SBM-based scheduling problems were solved using GAMS/CPLEX [34]. The dynamic optimization problems were reformulated as MINLPs using a full-discretization approach as described in the work by Flores-Tlacuahuac et al. [50]. The optimal solutions are shown in Tables 5.1-5.4. The optimal solution for SBM-based scheduling was validated via simulation on the closed-loop system using the derived input-output linearizing controller.

Table 5.3: Optimal Solution for Static Scheduling

k	1	2	3	4
τ_k	10	10	10	10
i	1	2	3	4
W_i	35.100	21.525	20.330	27.526
$tp_{i,k}$	$(i = 2, k = 1)$	$(i = 1, k = 2)$	$(i = 3, k = 3)$	$(i = 4, k = 4)$
	16.064	15.171	18.350	30

Table 5.4: Optimal Solution for Full Dynamic Scheduling

k	1	2	3	4
τ_k	10.937	10	10	11.06
i	1	2	3	4
W_i	35.100	22.323	21.083	27.594
$tp_{i,k}$	$(i = 4, k = 1)$	$(i = 2, k = 2)$	$(i = 1, k = 3)$	$(i = 3, k = 4)$
	16.659	30	15.733	18.396

5.4.1.2 Performance in the Presence of Model Uncertainty

The aforementioned results were obtained assuming no plant-model mismatch, i.e., that the dynamic model used is perfect. Here, we assume that the reaction rate constant has been overestimated in the model by 10% compared to the plant. The control actions computed using full dynamic scheduling were imposed on the mismatched plant. Clearly, in the absence of feedback control, most of the products are off-spec (Figure 5.2, left). This occurs because this formulation does not incorporate a feedback control system. Additionally, the process becomes unstable owing to a runaway reaction because product 3 and 4 are manufactured in an open-loop unstable region [50]. Then, we imposed the setpoint sequence computed from SBM-based scheduling on the closed-loop system using the input-output linearizing controller.

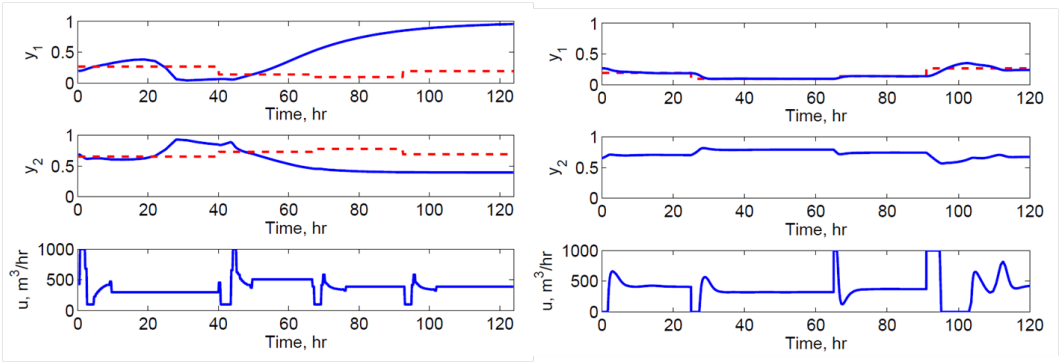

Figure 5.2: Process response using full dynamic (left) and SBM-based (right) scheduling in the presence of plant-model mismatch. Dashed lines represent the target values of the variables.

While each of the products is initially off-spec, feedback control with integral action compensates for plant-model mismatch and helps recover product purity (Figure 5.2, right).

5.4.2 Multi-product CSTR with External Heat Exchanger

We consider a more complex process network, comprising of a reactor and a heat exchanger shown in Figure 5.3. The dynamic process model and control problems have been formulated and solved in the work by Baldea et al. [7]. We rely on their modeling and control formulation, as well as model parameters from that paper.

5.4.2.1 Modeling of Process Network

The feed stream F contains the reactant A and its concentration is assumed to be constant. Two highly exothermic first-order reactions take

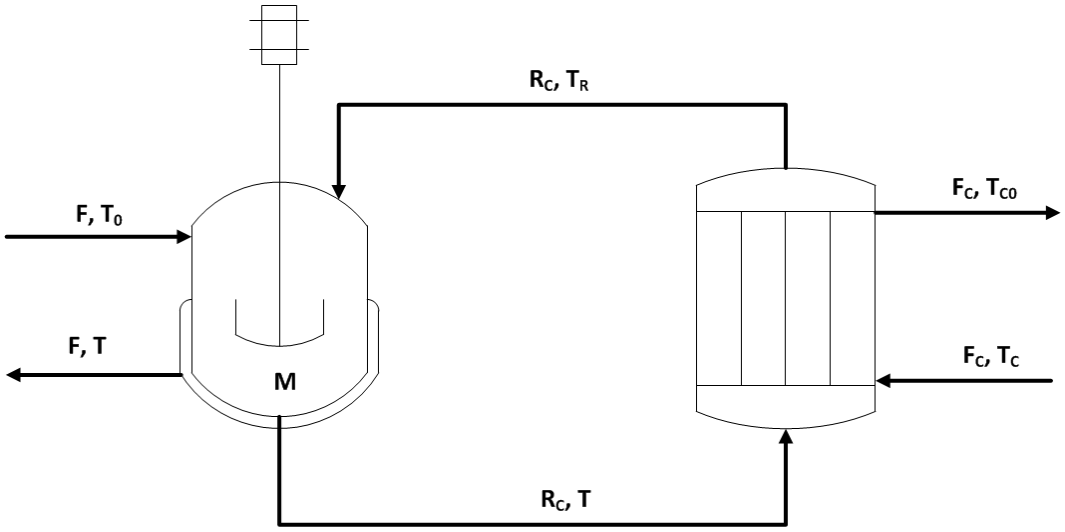

Figure 5.3: Schematic Diagram of a Process Network with External Heat Exchanger [7]

place, which produce products B and C (A $\xrightarrow{k_1}$ B $\xrightarrow{k_2}$ C). The large magnitudes of heat of reaction mean that the thermal effects of the reactions dominate, thus the adiabatic operation is not possible. In order to mitigate such highly exothermic reactions, the reaction mass is recycled at a higher rate compared to the feed through the external heat exchanger.

The following assumptions are made regarding the process:

- The flowrate, composition, and temperature of the feed remain constant.
- The inlet temperature of coolant remains constant.
- The reactions only occur inside the reactor.
- The heat capacity of the materials remain constant.

Under these assumptions, the material and energy balance equations for this system can be written as:

$$\frac{dC_A}{dt} = \frac{F}{M}(C_{A0} - C_A) - k_{10}e^{\frac{-E_{a1}}{RT}}C_A \tag{5.15}$$

$$\frac{dC_B}{dt} = -\frac{F}{M}C_B + k_{10}e^{\frac{-E_{a1}}{RT}}C_A - k_{20}e^{\frac{-E_{a2}}{RT}}C_B \tag{5.16}$$

$$\frac{dC_C}{dt} = -\frac{F}{M}C_C + k_{20}e^{\frac{-E_{a2}}{RT}}C_B \tag{5.17}$$

$$\begin{aligned}\frac{dT}{dt} &= \frac{F}{M}(T_0 - T) + \frac{R_C}{M}(T_R - T) \\ &\quad - \frac{\Delta H_1}{c_p}k_{10}e^{\frac{-E_{a1}}{RT}}C_A - \frac{\Delta H_2}{c_p}k_{20}e^{\frac{-E_{a2}}{RT}}C_B\end{aligned} \tag{5.18}$$

$$\frac{dT_R}{dt} = \frac{R_C}{M_R}(T - T_R) - \frac{UA}{c_p M_R}(T_R - T_C) \tag{5.19}$$

$$\frac{dT_C}{dt} = \frac{F_C}{M_C}(T_{C0} - T_C) + \frac{UA}{c_{pc}M_C}(T_C - T_R) \tag{5.20}$$

where C_A, C_B, and C_C are concentrations of A, B, and C. T is the temperature of the reactor, T_R is the temperature of reaction mass in the tube side of the heat exchanger, and T_C is the outlet temperature of the coolant. All the process parameters can be found in [7] and are reproduced in Table 5.5 for completeness.

In this continuous reactor, a number of product grades are specified in terms of concentrations. These products are produced from the same raw materials but at different operating conditions. As the reactor operates continuously, the switch from different products involves a dynamic transition, which is carried out by changing the composition setpoint.

Table 5.5: CSTR Model Parameters

Parameters	Values
F	$20l/min$
M	$1200l$
M_R	$22.93l$
M_C	$68.8l$
C_{A0}	$2mol/l$
U	$1987.5W/m^2\text{-}K$
A	$11.14m^2$
ΔH_1	$-791.2kJ/mol$
ΔH_2	$-527.5kJ/mol$
E_{a1}	$75.36kJ/mol$
E_{a2}	$150.72kJ/mol$
k_{10}	$5.35 \times 10^{10} min^{-1}$
k_{20}	$4.61 \times 10^{18} min^{-1}$
c_p	$4138.2J/l\text{-}K$
c_{pc}	$4138.2J/l\text{-}K$
T_0	$311.1K$
T_{C0}	$294.4K$

5.4.2.2 Control Strategy

Due to the differences in the magnitude of R_C and F, the model is stiff featuring a two-time scale behavior. The concentrations of the output stream takes a long time to evolve, while the temperatures of the system reach new steady state values quickly [7]. This two-time scale behavior poses challenges for controller design, and should be dealt with by deriving a low-order, non-stiff model of the slow system dynamics for model-based control purposes (see, e.g., the discussions in Kokotovic et al. and Baldea and Daoutidis) [64, 8].

Using singular perturbation arguments, separate models for the fast and slow components of the system dynamics can be defined [7]. A cascade control structure is used, in which the reactor temperature (fast dynamics) is controlled by a proportional-integral (PI) controller. The setpoint of the controllers in the fast time scale are available as manipulated inputs in the slow time scale. In order to track and control the purity of the product, a nonlinear controller is implemented. This drives the system so that C_B follows a first order linear dynamics, and relies on the setpoint of the temperature controller as a manipulated variable. The controllers are shown below [7].

$$F_C = F_{CS}\left(1 + K_C\big(T - T_{SP} + \frac{1}{\tau_I}\int_0^t (T - T_{SP})dt\big)\right) \tag{5.21}$$

$$C_{B,SP} = C_B + \gamma_1 \frac{dC_B}{dt} \tag{5.22}$$

$$\tag{5.23}$$

In Equation 5.22, the first term corresponds to requesting a first order response in C_B when using a standard input-output linearizing controller. The controller parameters K_C, τ_I, and γ_1 are chosen to be $0.15K^{-1}$, $2.8min$ and $30min$, respectively [7]. However, this would lead to closed-loop instability due to non-minimum phase behavior which comes from the increased second reaction rate, thus the auxiliary term is designed to cancel the influence of the second reaction [7].

5.4.2.3 Scheduling

In this continuous stirred reactor tank (CSTR), three different products are manufactured at three different operating conditions with the same raw materials. Since the reactor operates continuously, this requires the setpoint changes by adjusting the manipulated variable (MV), F_C. The detailed operating conditions for the products are shown in the Table below. The manufacturing of the products occurs in a production cycle, which consists of two different periods: the transition period and the production period. The transition period represents the dynamic transition between the products. The transition from one product to the other is considered complete when the controlled variable (CV), C_B, is within the tolerance (1.0×10^{-3}) of its new steady state setpoint. The production period represents the desired steady state operating condition. The operating conditions for all the products are shown in Table 5.6, which can be found in [7]. The production cycle is assumed to be cyclical.

Table 5.6: Operating Conditions of Each Product

Product	C_A	C_B	C_C	T	T_R	T_C	F_C
1	0.0194	1.550	0.434	375.02	346.24	323.18	257.16
2	0.0145	1.323	0.663	379.57	349.31	324.66	261.45
3	0.0103	0.995	0.995	385.14	353.26	326.28	271.77

Table 5.7: Product Information

Product	$C_P(\$/L)$	$C_I(\$/L\text{-}hr)$	$D_{LB}(L)$	$D_{UB}(L)$
1	0.9	0.066	10,000	40,000
2	1.0	0.060	10,000	40,000
3	1.1	0.054	10,000	40,000

The objective is to identify the optimal operating sequence that meets the product demands while reducing the inventory cost. The decision variables include the production sequence (discrete), the total cycle time, and the production time of each product. It was assumed that the product price, inventory cost, and the demand for each product is deterministic and known a priori. Also, it was assumed that only one product is produced in each time slot, and it is produced only once during the entire production cycle. All the product pricing information is included in Table 5.7.

In this case, we approach the scheduling problem using the full-order process model from a slightly different perspective, in the sense that the controller of the process is also included. That is, Equations 5.15-5.20 and the controller equations (Equations 5.21-5.22), are included as constraints, as shown in Equation 5.24.

$$
\begin{aligned}
\max_{T_C, T_j, b_i} \quad & \left(\sum_{i=1}^{N_P} C_{P,i} P_i - \sum_{i=1}^{N_P} C_{I,i} W_i \right) \\
\text{subject to} \quad & T_C = \sum_{j=1}^{N_S} T_j \\
& T_{S,LB} \leq T_j \leq T_{S,UB} \\
& T_{C,LB} \leq T_C \leq T_{C,UB} \\
& \sum_{i=1}^{N_P} b_i = 1 \qquad b_i \in [0,1] \\
& N_P = N_S \\
& P_i = \int_0^{T_C} F dt \qquad \text{if } C_B = C_{B,i} \\
& W_i = \int_0^{T_C} P_i dt \\
& D_{LB,i} \leq P_i \leq D_{UB,i} \\
& C_{B,SP} = \sum_{i=1}^{N_P} C_{B,i} b_i \\
& \text{Dynamic Process Model} \\
& \text{Control Law}
\end{aligned}
\tag{5.24}
$$

Conversely, the SBM-based scheduling problem is formulated as:

$$
\begin{aligned}
\max_{T_C, T_j, b_i} \quad & \left(\sum_{i=1}^{N_P} C_{P,i} P_i - \sum_{i=1}^{N_P} C_{I,i} W_i \right) \\
\text{subject to} \quad & T_C = \sum_{j=1}^{N_S} T_j \\
& T_{S,LB} \leq T_j \leq T_{S,UB} \\
& T_{C,LB} \leq T_C \leq T_{C,UB} \\
& \sum_{i=1}^{N_P} b_i = 1 \qquad b_i \in [0,1] \\
& N_P = N_S \\
& P_i = \int_0^{T_C} F dt \qquad \text{if } C_B = C_{B,i} \\
& W_i = \int_0^{T_C} P_i dt \\
& D_{LB,i} \leq P_i \leq D_{UB,i} \\
& C_{B,SP} = \sum_{i=1}^{N_P} C_{B,i} b_i \\
& \tau \frac{dC_B}{dt} + C_B = C_{B,SP}
\end{aligned}
\tag{5.25}
$$

in which, the DAE of the process model and the nonlinear control equations are replaced by a linear ordinary differential equation, as shown in Equation **??**.

5.5 Simulation Results

The optimal solutions for the full MIDO approach and the SBM MIDO approach are shown in Tables 5.8 - 5.9 and Figure 5.4. The results obtained using the two different methods are similar. The production times of all three

Table 5.8: Optimal Schedule Comparison

Method	Full MIDO	SBM MIDO
T_C	1800.28	1800
T_1	600.28	600
T_2	600	600
T_3	600	600
Sequence	$1 \rightarrow 3 \rightarrow 2 \rightarrow 1$	$1 \rightarrow 3 \rightarrow 2 \rightarrow 1$
Solution Time (s)	606	68

Table 5.9: Optimization Result

Method	Full MIDO	SBM MIDO
P_1	1.021×10^4	1.109×10^4
P_2	1.036×10^4	1.120×10^4
P_3	1.000×10^4	1.071×10^4
W_1	2.612×10^6	3.076×10^6
W_2	8.954×10^6	9.895×10^6
W_3	1.441×10^7	1.567×10^7
Profit	3.198361	3.104565

time slots are similar, and reflect the fact that the optimal solution corresponds to meeting the minimum product demand. Also, the production sequences of both methods are the same, where the process is initialized at the states of product 1. However, the comparison of the simulation times emphasizes the benefit of using the SBM formulation. It only takes about 60 seconds to solve the scheduling problem, while the full MIDO takes about 10 minutes to solve. The trajectories of the states and the manipulated variables are shown in the following Figure 5.4. The result was calculated using gPROMS [79]. The following values (Table 5.10) were used for the optimization.

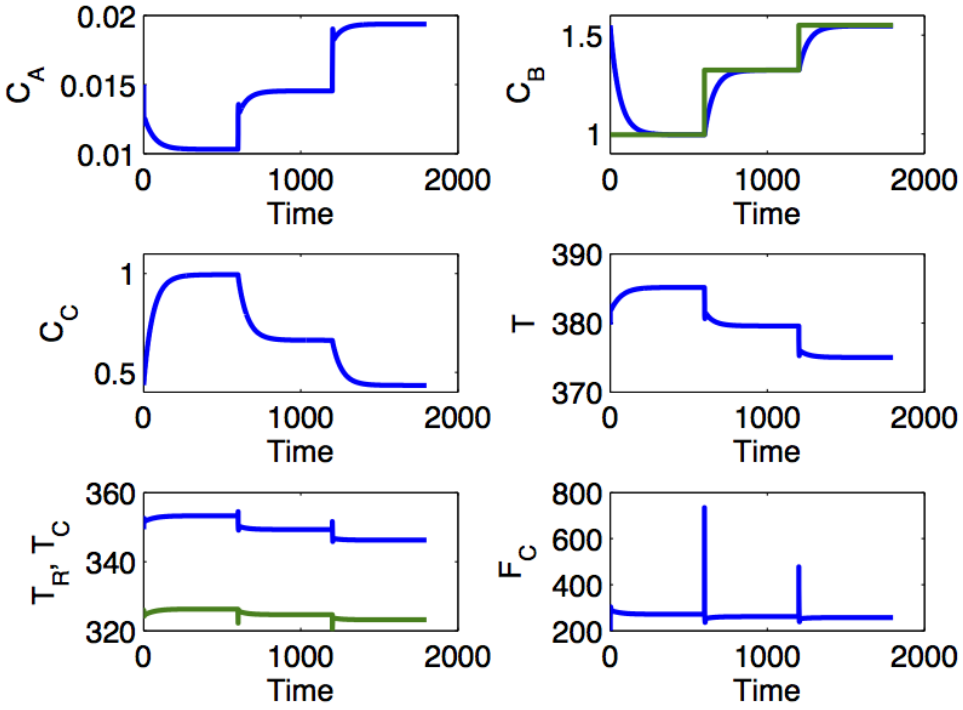

Figure 5.4: Optimal Dynamic Profile of Full MIDO

Table 5.10: Optimization Parameters

Parameters	Values
$T_{S,LB}$	400
$T_{S,UB}$	3,000
$T_{C,LB}$	1,200
$T_{C,UB}$	9,000

5.6 Summary

In this chapter, integration of scheduling and control is discussed. Existing solution approaches such as static scheduling and scheduling under full process constraints and their shortcomings are summarized. The key contribution of this chapter is the introduction of the scale bridging model (SBM) concept, as a low-order representation of the closed-loop dynamics of the process, which are relevant for scheduling calculations. We rely on concept from input-output linearization to derive such models for a class of square nonlinear processes with stable zero dynamics.

Two case studies, the short term scheduling problem of a multi-product CSTR and a multi-product process with an external heat exchanger, were used to demonstrate the superiority of our approach (in terms of significant reduction in computational burden) compared to conventional formulations of the integrated scheduling and control problem.

Chapter 6

Summary and Future Directions

6.1 Summary of Contributions

This dissertation focused on a specific class of processes, that combine both batch and continuous processing of materials. We refer to these systems as batch to continuous (B2C) processes. High dimensionality, physical complexity and the need to make real-time model-based decisions preclude (in many cases) the development of first-principles models for such processes. Motivated by this, we aimed to develop a data-driven framework for extracting useful analysis and control-relevant information from data collected during B2C process operations.

Specifically, we developed data pretreatment, modeling and optimization methods, and demonstrated applications on an industrial system. We note that the methods are described generally and can be extended to many other sequential batch-continuous processes in the food, plastics and ceramic manufacturing industries.

In Chapter 2, data cleaning methods such as outlier detection and filtering are discussed. There are several techniques to detect and remove outliers used in chemical industry. The most prevalent technique is to use the

3σ to determine the upper and lower bounds of an "acceptable" region. We compare this to the Hampel Identifier, which has superior performance. For noise filtering, we compare median-based and low-pass filters, as well as the Savitzky-Golay filter. We conclude that for the class of systems considered, the Hampel Identifier and Savitzky-Golay filter yield the best performance.

Chapter 2 also introduces a novel approach to correlate the variable-wise unfolded batch data to the subsequent continuous data. Most of the recent studies in modeling and analysis have focused solely on investigating batch and continuous process separately; however, sequential batch-continuous process is a hybrid system in which both of these modes of process coexist and affect the final product quality. In order to correlate these processes, a characteristic value of the batch is obtained and assigned until the following batch is introduced to the system. In this way, the variable-wise unfolded batch data has the same data length as the continuous data, the batch-to-batch variability is maintained, and it can be used for further analysis without any loss in information.

Chapter 3 focuses on the data-driven modeling aspect. Widespread data-driven modeling methods such as PCA and PLS are discussed in detail in this Chapter. Also, special variations of these methods such as kernel PCA/PLS which accounts for the nonlinearity and multiway PCA/PLS which focuses on the batch trajectory are addressed.

We first treat the batch and continuous sections of the process separately: i) PCA is applied to the upstream batch process, where distribution

of the raw materials is measured to represent the status of the batches. The Q-statistics and Hotelling's T^2 identify the variability within the captured principal components and errors outside the retained principal components, respectively. ii) Similar techniques are applied to the continuous process for data reduction and process monitoring. Finally, we consider the global monitoring of the B2C process. Here, in order to account for different operating modes, a clustering method such as k-means clustering is used to partition the original dataset into numerous similar subsets, which can also act as a reference point when implemented online to determine which mode the current operation belongs to. Chapter 3 also investigates variable selection methods for PLS in order to improve the accuracy of the model.

Chapter 4 introduces a method to align the offline quality variable to the online continuous process measurement. After the alignment, the soft sensor is developed to represent the product quality. This developed soft sensor can monitor and assess the current state of the production; however, a real benefit is achieved by optimizing the production. Chapter 4 uses real-time optimization (RTO) to calculate the optimal sequence of manipulated input that minimizes any deviation from the predefined set-point. Two different formulations (tight set-point tracking and tracking with a buffer) show that tight product quality control can be accomplished while reducing the toll on valve movements.

Lastly, Chapter 5 focuses on a broader topic, the integration of production scheduling and process control decisions, motivated by the need to

improve operating economic under, e.g., fast-changing market conditions. In order to alleviate the computational burden of embedding a (high-dimensional, nonlinear) dynamic model in a scheduling calculation, we propose a new approach based on scale-bridging models (SBMs). An SBM is the low-order representation of the scheduling-relevant *closed-loop* dynamics of a process. We demonstrate that one possible embodiment of the SBM concept is the use of input-output linearizing controllers, whose explicitly defined, linear input-output behavior can be used as an SBM for scheduling calculations. Two case studies, a simple multi-product CSTR and a more complex process network demonstrate that this approach is computationally favorable compared to using full-order, detailed process models in scheduling calculations, while leading to the same economic performance.

6.2 Potential Directions for Future Work

In this dissertation, a foundation for modeling sequential batch-continuous process has been established; however, there still remain a number of issues that need to be addressed. We have, in particular, focused on developing linear steady-state models for monitoring and product quality; however, in order to ensure stability and obtain higher economic and operational benefit, the model needs to include process dynamics (which is not only limited to the relationship between the manipulated variables and the controlled variables, but also extended to the relationship between the disturbance variables and the controlled variables). This task raises interesting issues in estimating time

delays in sequential batch-continuous process. Estimating time delays between the variables is not a simple task and remains an open research topic [129].

Also, different model forms can be considered. Data-driven modeling methods such as autoregressive with exogenous input or first-principles modeling methods have advantages and disadvantages. Data-driven modeling can be carried out to identify the order and the model parameters; however, the industrial system exhibits multiple operating modes, which need to be taken into account. This may lead to developing numerous models which will be difficult to implement and maintain. On the other hand, first-principles modeling methods may be able to handle multiple operating modes; however, identifying and fitting the model form to the given data may be too difficult.

The development of dynamic models can support the analysis and implementation of a feedforward/feedback controller. Given that each of the multiple operating modes may require a specific control law, a stability analysis must be carried out.

On the topic of integrating scheduling and control, the proposed SBM method should be expanded to incorporate the dynamics of batch and continuous processes of sequential batch-continuous process. The SBM has been shown to work with multi-input multi-output (MIMO) system, process with model predictive control (MPC), and autoregressive with exogenous terms (ARX) model [41, 13, 120]. These developments can be used to expand the application of the SBM to batch processes. The low(er)-order dynamic model can be used to represent the batch output, which can be used to schedule

batch cycles. This idea can be extended to incorporating both batch and continuous processes of the sequential batch-continuous process in the scheduling framework. Two sets of SBMs can be used in series to represent the process dynamics of the batch and continuous processes. Also, *rescheduling* scenarios can be considered, based, e.g., on the effect of process disturbances in the upstream batch process and the effect of process constraint changes in the downstream continuous process.

Lastly, in this dissertation, we focused on an industrial material processing system that deals with a two-phase suspension. In this particular industrial case study, our focus was on operating the system efficiently in terms of driving the system to manufacture on-spec products. More broad operating objectives (especially economic ones) should be considered. In particular, the output of a B2C process becomes the feed of an oftentimes energy intensive downstream process (e.g., kiln drying, furnace firing). The quality parameters of the B2C product will impact the energy use of the downstream process, and carrying out an "overall" optimization of the entire system, with the purpose of minimizing overall energy consumption is a likely worthwhile goal.

Bibliography

[1] Z. H. Abu-el-zeet, V. M. Becerra, and P. D. Roberts. Combined bias and outlier identification in dynamic data reconciliation. *Computers & Chemical Engineering*, 26:921–935, 2002.

[2] R. J. Allgor and P. I. Barton. Mixed-integer dynamic optimization I: Problem formulation. *Computers & Chemical Engineering*, 23(4):567–584, 1999.

[3] R. Amrit and J. B. Rawlings. Economic optimization using model predictive control with a terminal cost. *Annual Review in Control*, 35:178–186, 2011.

[4] C. M. Andersen and R. Bro. Variable selection in regression-a tutorial. *Journal of Chemometrics*, 24:728–737, 2010.

[5] M. Andersson. A comparison of nine PLS1 algorithms. *Journal of Chemometrics*, 23:518–519, 2009.

[6] S. Bahroun, S. Li, C. Jallut, C. Valentin, and F. De Panthou. Control and optimization of a three-phase catalytic slurry intensified continuous chemical reactor. *Journal of Process Control*, 20:664–675, 2010.

[7] M. Baldea and P. Daoutidis. Model reduction and control of reactor–heat exchanger networks. *Journal of Process Control*, 16:265–274, 2006.

[8] M. Baldea and P. Daoutidis. *Dynamics and Nonlinear Control of Integrated Process Systems*. Cambridge University Press, Cambridge, 2012.

[9] M. Baldea, P. Daoutidis, and A. Kumar. Dynamics and control of integrated networks with purge streams. *AIChE Journal*, 52:1460–1472, 2006.

[10] M. Baldea, N. H. El-Farrac, and B. E. Ydstie. Dynamics and control of chemical process networks: Integrating physics, communication and computation. *Computers & Chemical Engineering*, 51:43–54, 2013.

[11] M. Baldea and I. Harjunkoski. Integrated production scheduling and process control: A systematic review. *Computers & Chemical Engineering*, 71:377–390, 2014.

[12] M. Baldea and C. R. Touretzky. Nonlinear model predictive control of energy-integrated process systems. *Systems & Control Letters*, 62:723–731, 2013.

[13] Michael Baldea, Juan Du, Jungup Park, and Iiro Harjunkoski. Integrated production scheduling and model predictive control of continuous processes. *AIChE Journal*, 61:4179–4190, 2015.

[14] V. Barnett and T. Lewis. *Outliers in Statistical Data*. John Wiley & Sons, Hoboken, NJ, 1994.

[15] L. Biegler. An overview of simultaneous strategies for dynamic optimization. *Chemical Engineering and Processing*, 46:1043–1053, 2007.

[16] G. Birol, C. Undey, and A. Cinar. A modular simulation package for fed-batch fermentation: Penicillin production. *Computers & Chemical Engineering*, 26:1553–1565, 2002.

[17] W. Cai, Y. Li, and X. Shao. A variable selection method based on uninformative variable elimination for multivariate calibration of near-infrared spectra. *Chemometrics and Intelligent Laboratory Systems*, 90:188–194, 2008.

[18] P. Cassagnau, V. Bounor-Legaré, and F. Fenouillot. Reactive processing of thermoplastic polymers: A review of the fundamental aspects. *International Polymer Processing*, 22:218–258, 2007.

[19] P. M. Castro and I. E. Grossmann. New continuous-time MILP model for the short-term scheduling of multistage batch plants. *Industrial & Engineering Chemistry Research*, 44:9175–9190, 2005.

[20] V. Centner, D. Massart, O. E. De Noord, S. De Jong, B. M. Vandeginste, and C. Sterna. Elimination of uninformative variables for multivariate calibration. *Analytical Chemistry*, 68:3851–3858, 1996.

[21] C. Chatzidoukas, C. Kiparissides, J. D. Perkins, and E. N. Pistikopoulos. Optimal grade transition campaign scheduling in a gas-phase polyolefin fbr using mixed integer dynamic optimization. *Computer Aided Chemical Engineering*, 14:71–76, 2003.

[22] J. Chen, A. Bandoni, and J. A. Romagnoli. Outlier detection in process plant data. *Computers & Chemical Engineering*, 22:641–646, 1998.

[23] J. Chen and C. Liao. Dynamic process fault monitoring based on neural network and PCA. *Journal of Process Control*, 12:277–289, 2002.

[24] J. Chen, C. Liao, F. R. J. Lin, and M. Lu. Principle component analysis based control charts with memory effect for process monitoring. *Industrial & Engineering Chemistry Research*, 40:1516–1527, 2001.

[25] J. Chen and J. A. Romagnoli. A strategy for simultaneous dynamic data reconciliation and outlier detection. *Computers & Chemical Engineering*, 22:559–562, 1998.

[26] T. Chen, J. Morris, and E. Martin. Dynamic data rectification using particle filters. *Computers & Chemical Engineering*, 32:451–462, 2008.

[27] L. H. Chiang, R. D. Braatz, and E. L. Russell. *Fault Detection and Diagnosis in Industrial Systems (Advanced Textbooks in Control and Signal Processing)*. Springer, 2001.

[28] L. H. Chiang, R. J. Pell, and M. B. Seasholtz. Exploring process data with the use of robust outlier detection algorithms. *Journal of Process Control*, 13:437–449, 2003.

[29] J. Cho, J. Lee, S. W. Choi, D. Lee, and I. Lee. Fault identification for process monitoring using kernel principal component analysis. *Chemical Engineering Science*, 60:279–288, 2005.

[30] S. W. Choi, C. Lee, J. Lee, J. H. Park, and I. Lee. Fault detection and identification of nonlinear processes based on kernel PCA. *Chemometrics and Intelligent Laboratory Systems*, 75:55–67, 2005.

[31] I. Chong and C. Jun. Performance of some variable selection methods when multicollinearity is present. *Chemometrics and Intelligent Laboratory Systems*, 78:103–112, 2005.

[32] Y. Chu and F. You. Integration of scheduling and control with online closed-loop implementation: fast computational strategy and a large-scale global optimization algorithm. *Computers & Chemical Engineering*, 47:248–268, 2012.

[33] M. Contou-Carrere, M. Baldea, and P. Daoutidis. Dynamic precompensation and output feedback control of integrated process networks. *Industrial & Engineering Chemistry Research*, 43:3528–3538, 2004.

[34] CPLEX. Available at http://www.gams.com/dd/docs/solvers/cplex/. Accessed on March 8, 2014.

[35] P. Daoutidis and C. Kravaris. Dynamic output feedback control of minimum-phase nonlinear processes. *Chemical Engineering Science*, 4:837–849, 1992.

[36] P. Daoutidis and C. Kravaris. Dynamic output feedback control of minimum-phase multivariable nonlinear processes. *Chemical Engineering Science*, 49:433–447, 1994.

[37] P. Daoutidis, M. Soroush, and C. Kravaris. Feedforward/Feedback control of multivariable nonlinear processes. *AIChE Journal*, 36:1471–1484, 1990.

[38] M. Daszykowski, K. Kaczmarek, Y. Vander Heyden, and B. Walczak. Robust statistics in data analysis a review: Basic concepts. *Chemometrics and Intelligent Laboratory Systems*, 85:203–219, 2007.

[39] S. De Jong. SIMPLS: An alternative approach to partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 18:251–263, 1993.

[40] G. Diana and C. Tommasi. Cross-validation methods in principal component analysis: A Comparison. *Statistical Methods & Applications*, 11:71–82, 2002.

[41] Juan Du, Jungup Park, Iiro Harjunkoski, and Michael Baldea. A time scale-bridging approach for integrating production scheduling and process control. *Computers & Chemical Engineering*, 79:59–69, 2015.

[42] R. Dunia, T. F. Edgar, T. Blevins, and W. Wojsznis. Multistate analytics for continuous processes. *Journal of Process Control*, 22:1445–1456, 2012.

[43] R. Dunia, T. F. Edgar, and M. Nixon. Process monitoring using principal components in parallel coordinates. *AIChE Journal*, 59:445–456, 2012.

[44] R. Dunia, V. Kumar, T. F. Edgar, T. Blevins, and W. Wojsznis. Multistate PCA for continuous processes. In *American Control Conference*, 2012.

[45] R. Dunia, S. J. Qin, T. F. Edgar, and T. J. McAvoy. Identification of faulty sensors using principal component analysis. *AIChE Journal*, 42:2797–2812, 1996.

[46] S. Engell. Feedback control for optimal process operation. *Journal of Process Control*, 17:203–219, 2007.

[47] S. Engell. Online optimizing control: The link between plant economics and process control. *Computer Aided Chemical Engineering*, 27:79–86, 2009.

[48] J. A. Fernández Pierna, O. Abbas, V. Baeten, and P. Dardenne. A backward variable selection method for PLS regression (BVSPLS). *Analytical Chimica Acta*, 642:89–93, 2009.

[49] J. Fish. *Multiscale Methods: Bridging the Scales in Science and Engineering*. Oxford University Press, Oxford, 2009.

[50] A. Flores-Tlacuahuac and I. E. Grossmann. Simultaneous cyclic scheduling and control of a multiproduct CSTR. *Industrial & Engineering Chemistry Research*, 45:6698–6712, 2006.

[51] A. Flores-Tlacuahuac and I. E. Grossmann. Simultaneous cyclic scheduling and control of a tubular reactors: Parallel production lines. *Industrial & Engineering Chemistry Research*, 50:8086–8096, 2011.

[52] I. E. Grossmann. Enterprise-wide optimization: A new frontier in process systems engineering. *AIChE Journal*, 51:1846–1857, 2005.

[53] F. E. Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 11:1–21, 1969.

[54] M. A. Gutierrez-Limón and A. Flores-Tlacuahuac. A scheduling and nonlinear predictive control strategy for continuous polymerization reactors. *Macromolecular Reaction Engineering*, 8:347–357, 2014.

[55] I. Harjunkoski, R. Nyström, and A. Horch. Integration of scheduling and control - theory of practice? *Computers & Chemical Engineering*, 33(12):1909–1918, 2009.

[56] M. Heidarinejad, J. , and P. D. Christofides. Economic model predictive control of nonlinear process systems using Lyapunov techniques. *AIChE Journal*, 58:855–870, 2011.

[57] A. Hoskuldsson. PLS regression methods. *Journal of Chemometrics*, 2:211–228, 1988.

[58] M. G. Ierapetritou and C. A. Floudas. Effective continuous-time formulation for short-term scheduling. 1. Multipurpose batch processes. *Industrial & Engineering Chemistry Research*, 37:4341–4359, 1998.

[59] J. E. Jackson. *A Users Guide To Principal Components*. John Wiley & Sons, Hoboken, NJ, 1991.

[60] J. E. Jackson and G. S. Mudholkar. Control procedures for residuals associated with principal component analysis. *Technometrics*, 21:341–349, 1979.

[61] S. S. Jogwar, M. Baldea, and P. Daoutidis. Tight energy integration: Dynamic impact and control advantages. *Computers & Chemical Engineering*, 34:1457–1466, 2010.

[62] J. V. Kadam and W. Marquardt. Integration of economical optimization and control for intentionally transient process operation. *Assessment and Future Directions of Nonlinear Model Predictive Control*, 358:419–434, 2007.

[63] P. Kadlec, B. Gabrys, and S. Strandt. Data-driven soft sensors in the process industry. *Computers & Chemical Engineering*, 33:795–814, 2009.

[64] P. V. Kokotovic, H. K. Khalil, and J. O'Reilly. *Singular Perturbations in Control: Analysis and Design*. Academic Press, London, 1986.

[65] S. Kolavennu, S. Palanki, and J. C. Cockburn. Nonlinear control of non-square multivariable systems. *Chemical Engineering Science*, 56:2103–2110, 2001.

[66] T. Kourti. Multivariate dynamic data modeling for analysis and statistical process control of batch processes, start-ups and grade transitions. *Journal of Chemometrics*, 17:93–109, 2003.

[67] T. Kourti and J. F. MacGregor. Process analysis, monitoring and diagnosis, using multivariate projection methods. *Chemometrics and Intelligent Laboratory Systems*, 28:3–21, 1995.

[68] N. Kramer and M. Sugiyama. The degrees of freedom of partial least squares regression. *Journal of the American Statistical Association*, 106:697–705, 2011.

[69] C. Kravaris and J. C. Kantor. Geometric methods for nonlinear process control. 1. Background. *Industrial & Engineering Chemistry Research*, 29:2295–2310, 1990.

[70] C. Kravaris and J. C. Kantor. Geometric methods for nonlinear process control. 2. Controller synthesis. *Industrial & Engineering Chemistry Research*, 29:2310–2323, 1990.

[71] J. V. Kresta, J. F. MacGregor, and T. E. Marlin. Multivariate statistical monitoring of process operating performance. *Canadian Journal of Chemical Engineering*, 69:35–47, 1991.

[72] W. Ku, R. H. Storer, and C. Georgakis. Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 30:179–196, 1995.

[73] A. Kumar and P. Daoutidis. Modeling, analysis and control of ethylene glycol reactive distillation column. *AIChE Journal*, 45:51–68, 1999.

[74] R. Leardi and A. L. González. Genetic algorithms applied to feature selection in PLS regression: how and when to use them. *Chemometrics and Intelligent Laboratory Systems*, 41:195–207, 1998.

[75] D. S. Lee and P. A. Vanrolleghem. Monitoring of a sequencing batch reactor using adaptive multiblock principal component analysis. *Biotechnology and Bioengineering*, 82:489–497, 2003.

[76] D. S. Lee and P. A. Vanrolleghem. Adaptive consensus principal component analysis for on-line batch process monitoring. *Environmental Monitoring and Assessment*, 92:119–135, 2004.

[77] J. Lee, C. Yoo, S. W. Choi, P. A. Vanrolleghem, and I. Lee. Nonlinear process monitoring using kernel principal component analysis. *Chemical Engineering Science*, 59:223–234, 2004.

[78] W. Li and S. J. Qin. Consistent dynamic PCA based on errors-in-variables subspace identification. *Journal of Process Control*, 11:661–678, 2001.

[79] Process Systems Enterprise Limited. gPROMS ModelBuilder 3.6.0, 2012.

[80] B. Lin, B. Recke, J. K. H. Knudsen, and S. B. Jørgensen. A systematic approach for soft sensor development. *Computers & Chemical Engineering*, 31:419–425, 2007.

[81] H. Liu, S. Shah, and W. Jiang. On-line outlier detection and data cleaning. *Computers & Chemical Engineering*, 28:1635–1647, 2004.

[82] J. Liu, D. Djurdjanovic, K. Marko, and J. Ni. Growing structure multiple model systems for anomaly detection and fault diagnosis. *Journal of Dynamic Systems, Measurement, and Control*, 131:051001, 2009.

[83] C. A. Lowry, W. H. Woodhall, C. W. Champ, and S. E. Rigdon. A multivariate exponentially weighted moving average control chart. *Technometrics*, 34:46–48, 1992.

[84] B. Lu, I. Castillo, L. Chiang, and T. F. Edgar. Industrial PLS model variable selection using moving window variable importance in projection. *Chemometrics and Intelligent Laboratory Systems*, 135:90–109, 2014.

[85] B. Lu and T. F. Edgar. *Improving process monitoring and modeling of batch-type plasma etching tools*. PhD thesis, The University of Texas at Austin, 2015.

[86] J. F. MacGregor and T. Kourti. Statistical process control of multivariate processes. *Control Engineering Practice*, 3:403–414, 1995.

[87] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, 1967.

[88] M. Maestri, A. Farall, P. Groisman, M. Cassanello, and G. Horowitz. A robust clustering method for detection of abnormal situations in a process with multiple steady-state operation modes. *Computers & Chemical Engineering*, 34:223–231, 2010.

[89] R. Mahadevan, F. J. Doyle III, and A. C. Allcock. Control-relevant scheduling of polymer grade transitions. *AIChE Journal*, 48(8):1754–1764, 2002.

[90] T. Mehmood, K. H. Liland, L. Snipen, and S. Sæ bø. A review of variable selection methods in partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 118:62–69, 2012.

[91] C. A. Méndez, J. Cerdá, I. E. Grossmann, I. Harjunkoski, and M. Fahl. State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Computers & Chemical Engineering*, 30:913–946, 2006.

[92] K. Mira, R. D. Gudi, S. C. Patwardhan, and G. Sardar. Resiliency issues in integration of scheduling and control. *Industrial & Engineering Chemistry Research*, 49(1):222–235, 2004.

[93] G. E. Moore. Cramming more components onto integrated circuits. In *Electronics*, pages 114–117, 1965.

[94] P. Nomikos and J. F. Macgregor. Monitoring batch processes using multiway principal component analysis. *AIChE Journal*, 40:13611375, 1994.

[95] P. Nomikos and J. F. Macgregor. Multi-way partial least squares in monitoring batch processes. *Chemometrics and Intelligent Laboratory Systems*, 30:97–108, 1995.

[96] P. Nomikos and J. F. Macgregor. Multivariate SPC charts for monitoring batch processes. *Technometrics*, 37:41–59, 1995.

[97] S. J. Orfanidis. *Introduction to Signal Processing*. Prentice Hall, Upper Saddle River, NJ, 1996.

[98] E. Ozel and S. Kurama. Effect of the processing on the production of cordierite-mullite composite. *Ceramics International*, 36:1033–1039, 2010.

[99] Jungup Park, Juan Du, Iiro Harjunkoski, and Michael Baldea. Integration of scheduling and control using internal coupling models. In *24th European Symposium on Computer Aided Process Engineering, PTS A and B, Book Series: Computer-Aided Chemical Engineering*, volume 33, pages 529–534, 2014.

[100] R. K. Pearson. Outliers in process modeling and identification. *IEEE Transactions on Control Systems Technology*, 10:55–63, 2002.

[101] R. K. Pearson. *Mining imperfect data: Dealing with contamination and incomplete records*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2005.

[102] J. M. Pinto and I. E. Grossmann. Optimal cyclic scheduling of multistage continuous multiproduct plants. *Computers & Chemical Engineering*, 18(9):797–816, 1994.

[103] G. Pison and S. Van Aelst. Analyzing data with robust multivariate methods and diagnostic plots. In *Compstat: Proceedings in Computational Statistics*, pages 165–170, 2002.

[104] A. Prata, J. Oldenburg, A. Kroll, and W. Marquardt. Integrated scheduling and dynamic optimization of grade transitions for a continuous polymerization reactor. *Computers & Chemical Engineering*, 32(3):463–476, 2008.

[105] S. J. Qin. Recursive PLS algorithms for adaptive data modeling. *Computers & Chemical Engineering*, 22:503–514, 1998.

[106] S. J. Qin. Survey on data-driven industrial process monitoring and diagnosis. *Annual Reviews in Control*, 36:220–234, 2012.

[107] J. S. Reed. *Principles of Ceramic Processing*. John Wiley & Sons, Hoboken, NJ, 1995.

[108] V. Romero Segovia, T. Hägglund, and K. J. Åström. Measurement noise filtering for PID controllers. *Journal of Process Control*, 24:299–313, 2014.

[109] R. Rosipal and N. Kramer. Overview and recent advances in partial least squares. *Subspace, Latent Structure and Feature Selection*, pages 34–51, 2006.

[110] E. L. Russell, L. H. Chiang, and R. D. Braatz. Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 51:81–93, 2000.

[111] T. L. M. Santos, P. E. A. Botura, and J. E. Normey-Rico. Dealing with noise in unstable dead-time process control. *Journal of Process Control*, 20:840–847, 2010.

[112] A. Savitzky and M. J. E. Golay. Smoothing and differentiation of data by simplified least squared procedures. *Analytical Chemistry*, 36:1627–1639, 1964.

[113] B. Schölkopf, A. Smola, and K. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Technical Report 44, Max-Planck-Institut fur biologische Kybernetik*, 1996.

[114] D. E. Seborg, T. F. Edgar, D. A. Mellichamp, and F. J. Doyle III. *Process Dynamics and Control*. John Wiley & Sons, Hoboken, NJ, 2011.

[115] M. N. Shirazi, H. Noda, and H. Sawai. A modular realization of adaptive PCA. *In International Conference on Systems, Man, and Cybernetics*, 4:3053–3056, 1997.

[116] D. E. Shobrys and D. C. White. Planning, scheduling and control systems: why cannot they work together. *Computers & Chemical Engineering*, 26(2):149–160, 2002.

[117] A. Singh. Outliers and robust procedures in some chemometric applications. *Chemometrics and Intelligent Laboratory Systems*, 33:75–100, 1996.

[118] A. Studart, U. T. Gonzenbach, E. Tervoort, and L. J. Gauckler. Processing routes to macroporous ceramics: A review. *Journal of the American Ceramic Society*, 89:1771–1789, 2006.

[119] S. Terrazas-Moreno, A. Flores-Tlacuahuac, and I. E. Grosmann. Simultaneous cyclic scheduling and optimal control of polymerization reactors. *AIChE Journal*, 53(9):2301–2315, 2007.

[120] C. Touretzky, T. Johansson, R. Pattison, I. Harjunkoski, and M. Baldea. Integrated scheduling and dynamic optimization of a cryogenic air separation unit subject to time-varying electricity prices. In *AIChE Annual Meeting*, 2015.

[121] C. R. Touretzky and M. Baldea. Integrating scheduling and control for economic MPC of buildings with energy storage. *Journal of Process*

Control, 24:1292–1300, 2014.

[122] C. R. Touretzky and M. Baldea. Nonlinear model reduction and model predictive control of residential buildings with energy recovery. *Journal of Process Control*, 24:723–739, 2014.

[123] V. S. Vassiliadis, R. W. H. Sargent, and C. C. Pantelides. Solution of a class of multistage dynamic optimization problems. 1. Problems with path constraints. *Industrial & Engineering Chemistry Research*, 33:2123–2133, 1994.

[124] V. S. Vassiliadis, R. W. H. Sargent, and C. C. Pantelides. Solution of a class of multistage dynamic optimization problems. 1. Problems without path constraints. *Industrial & Engineering Chemistry Research*, 33:2111–2122, 1994.

[125] Z. X. Wang, Q. P. He, and J. Wang. Comparison of variable selection methods for PLS-based soft sensor modeling. *Journal of Process Control*, 26:56–72, 2015.

[126] M. J. Watson, A. Liakopoulos, D. Brzakovic, and C. Georgakis. A practical assessment of process data compression techniques. *Industrial & Engineering Chemistry Research*, 37:267–274, 1998.

[127] S. Wold, P. Geladi, K. Esbensen, and Ohman J. Multi-way principal components-and PLS-analysis. *Journal of Chemometrics*, 1:41–56, 1987.

[128] S. Wold, M. Sjöström, and L. Eriksson. PLS-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 58:109–130, 2001.

[129] S. Xu, B. Lu, M. Baldea, T. F. Edgar, W. Wojsznis, T. Blevins, and M. Nixon. Data cleaning in the process industries. *Reviews in Chemical Engineering*, 31:453–490, 2015.

[130] H. H. Yue and S. J. Qin. Reconstruction-based fault identification using a combined index. *Industrial & Engineering Chemistry Research*, 40:4403–4414, 2001.

[131] Y. Zhang and T. F. Edgar. Bio-reactor monitoring with multiway-PCA and model based-PCA. In *American Institute of Chemical Engineers Conference on Computing and Systems Technology Division*, 2007.

[132] Y. Zhang and T. F. Edgar. On-line batch process monitoring using modified dynamic batch PCA. In *Proceedings of the American Control Conference*, 2007.

[133] Y. Zhang and T. F. Edgar. *Improved Methods in Statistical and First Principles Modeling for Batch Process Control and Monitoring.* PhD thesis, The University of Texas at Austin, 2008.

Vita

Jungup Park was born in Daegu, the Republic of Korea in 1988, the son of Dr. Sung Min Park and Dr. Geum Joo Han. He moved to the United States of America in 2003, where he attended high school in Indianapolis, IN. He received the Bachelor of Science degree in Chemical Engineering and Biomedical Engineering from the Carnegie Mellon University in 2011. During his time, he participated in active researches with Dr. Myung S. Jhon and Dr. Erik Ydstie. He received summer undergraduate research fellowship to continue his research in the summer of 2010. After obtaining his undergraduate degrees, he applied to the University of Texas at Austin for enrollment in their chemical engineering program. He was accepted and started his graduate studies in August, 2011. He continued his studies under the supervisions of Dr. Thomas F. Edgar and Dr. Michael Baldea.

Permanent address: 200 E Dean Keeton St. Stop C0400
Austin, Texas 78712-1589

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.